Exploring continual learning strategies in artificial neural networks through graph-based analysis of connectivity: insights from a brain-inspired perspective

Lucrezia Carboni, Dwight Nwaigwe, Marion Mainsant, Raphaël Bayle, Marina Reyboz, Martial Mermillod, Michel Dojat, Sophie Achard

Exploring Continual Learning Strategies in Artificial Neural Networks through Graph-Based Analysis of Connectivity: Insights from a Brain-Inspired Perspective

Lucrezia Carboni[a,b,1], Dwight Nwaigwe[a,b], Marion Mainsant[c,d], Raphael Bayle[d], Marina Reyboz[d], Martial Mermillod[c], Michel Dojat[b,*] and Sophie Achard[a]

[a] Univ. Grenoble Alpes, Inria, CNRS, INP, LJK, , Grenoble, 38000, France
[b] Univ. Grenoble Alpes, Inserm, U1216, Grenoble Institut Neurosciences, GIN, Grenoble, 38000, France
[c] Univ. Grenoble Alpes, Univ. Savoie Mont Blanc, CNRS, LPNC, Grenoble, 38000, France
[d] Univ. Grenoble Alpes, CEA, LIST, Grenoble, 38000, France

ARTICLE INFO

Keywords:
Brain connectivity
Graph theory
Catastrophic forgetting
Deep learning
Artificial intelligence
Incremental learning

ABSTRACT

Artificial Neural Networks (ANNs) aim at mimicking information processing in biological networks. In cognitive neuroscience, graph modeling is a powerful framework widely used to study brain structural and functional connectivity. Yet, the extension of graph modeling to ANNs has been poorly explored especially in term of functional connectivity (i.e. the contextual change of the activity's units in networks). From the perspective of designing more robust and interpretable ANNs, we study how a brain-inspired graph-based approach can be extended and used to investigate their properties and behaviors. We focus our study on different continual learning strategies inspired by the human brain and modeled with ANNs. We show that graph modeling offers a simple and elegant framework to deeply investigate ANNs, compare their performances and explore deleterious behaviors such as catastrophic forgetting.

1. Introduction

Artificial neural networks (ANNs) have been developed with the goal of mimicking the way biological neural mechanisms process information, learn, and make decisions (Rosenblatt [1958], LeCun et al. [2015], Parhi and Unnikrishnan [2020], Botvinick et al. [2020], Hassabis et al. [2017]). Since their development and over the years, there has been an interest in using insights from cognitive neuroscience to improve the performance of ANNs (McCulloch and Pitts [1943], McClelland and Rumelhart [1986], Hebb [2005], Marblestone et al. [2016], Hassabis et al. [2017], Khacef et al. [2018]). The use of brain-inspired models, such as spiking neural networks (SNN) and convolutional neural networks (CNN), has led to the development of innovative ANNs architectures that reproduce brain activation at the cellular level for SNN (Maass [1997]), at the brain level for CNN (Yamins et al. [2014], Cichy et al. [2016], Yamins and DiCarlo [2016], Kuzovkin et al. [2018]) or even control the cortical processing of visual information (Bashivan et al. [2019]). However, despite the seminal bio-inspiration in ANNs design, many tools introduced for brain information processing investigation have not been fully used to study ANN behavior. Among these, graph modeling has shown to be a powerful framework widely used in neuroscience to study brain structural and functional connectivity (Petersen and Sporns [2015], Bullmore and Sporns [2009], Wang et al. [2010], Sporns [2022], Barabási [2013]). In this paper, we propose to explore how such a framework can be elegantly and interestingly used to investigate the ANN properties and particular behaviors. A conceptual visualization of our proposal can be found in Fig. 1.

As a case of study, we concentrate on the sequential learning process where ANNs are trained on an ordered series of tasks as shown in Fig. 2 (Buzzega et al. [2020], Hadsell et al. [2020a,b]). This process mimics how the brain continuously learns and adapts to new tasks (Milgram et al. [1987], Pascual-Leone et al. [2005]). As neuroscientific literature reports, brain connectivity changes are associated with new learning tasks (Casimo [2018], de Vico Fallani et al. [2010], Zouridakis et al. [2007]), we explore, using graph modeling, the corresponding ANN connectivity changes. In particular, we investigate how specific graph statistics are modified during a continual learning framework and the conditions that lead to catastrophic forgetting (i.e. the performance of previously learned tasks dramatically decreases when new tasks are learned in a sequential manner by stopping training on task A while beginning training on task B, Fig. 2 Panel B).

In order to counter catastrophic forgetting, recent articles have proposed Dream Net, a brain-inspired ANN, basically simulating synaptic consolidation (Mainsant et al. [2021], Solinas et al. [2021]). This bio-inspired method is efficient but, more importantly, compared to other models, does

* Supported in part by MIAI@Grenoble Alpes (ANR 19-P3IA-003).

In this work we apply a brain connectivity-inspired model to characterize hidden units in an artificial neural network in a continual learning framework.

*Corresponding author
michel.dojat@inserm.fr (M. Dojat)
https://neurosciences.univ-grenoble-alpes.fr//fr/annuaire/michel-dojat--621497.kjsp (M. Dojat)
ORCID(s): 0000-0003-3519-3482 (L. Carboni); 0000-0003-2747-6845 (M. Dojat)

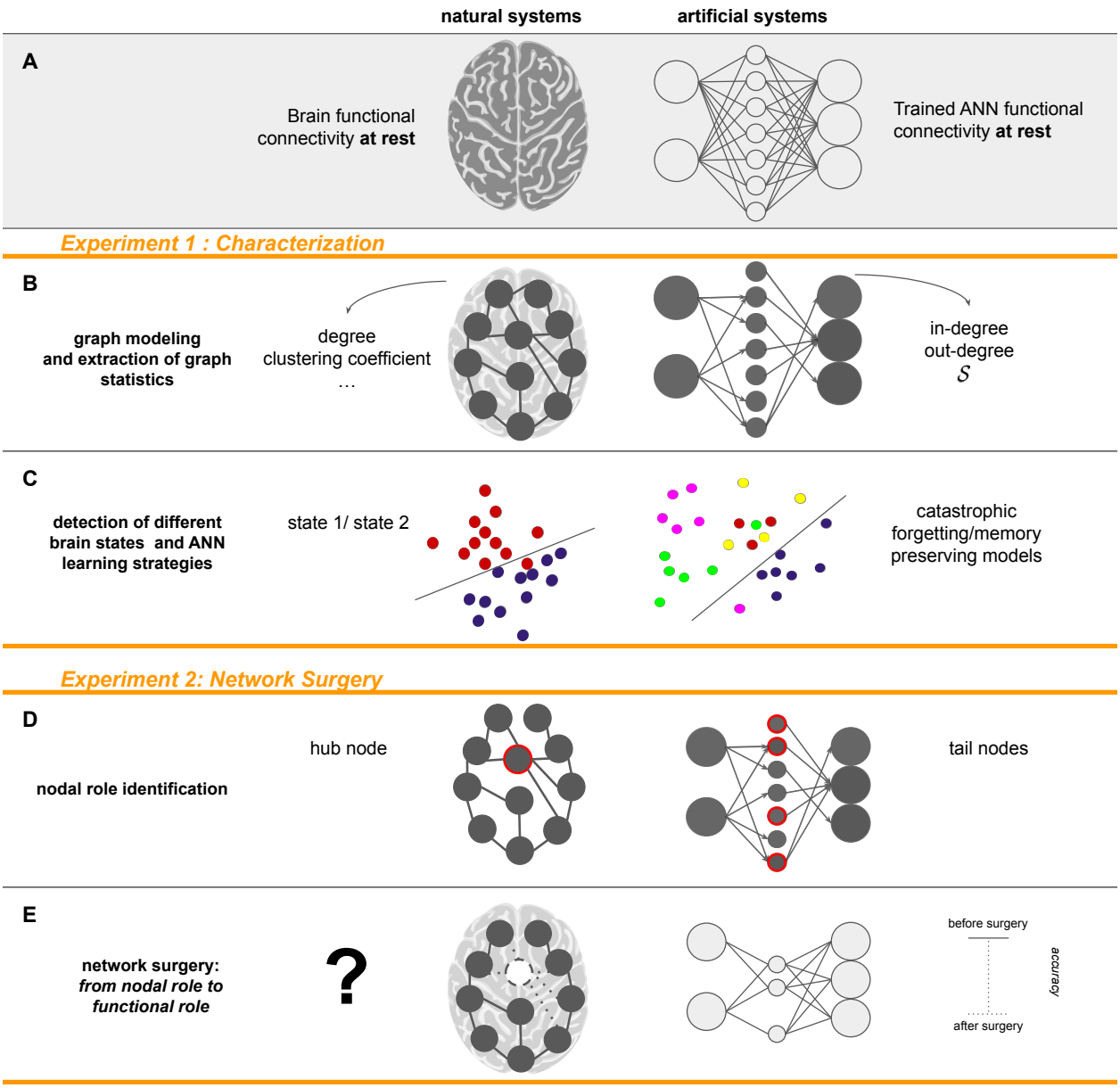


Figure 1: Overview of the brain-inspired perspective of the proposed graph-based analysis of ANNs. (A) Inspired by the concept of brain functional connectivity at rest, we propose to investigate the equivalent functional connectivity at rest of trained artificial systems. (B) Graph modeling can be applied for both brain and ANN connectivity modeling. Thus, different graph statistics can be extracted to characterize the resting state network. (C) Similarly to brain functional connectivity studies aiming at differentiating between different brain states (for instance healthy vs pathological conditions), by the use of the considered graph statistics, we show how the graph statistics can be used as features for the identification of ANN in good learning conditions or affected by the catastrophic forgetting phenomenon, or even to identify the learning strategy used. (D) At a finer level of a single-state characterization, nodal roles are investigated to determine the presence of nodes having specific roles in the network. In natural systems, nodal role discovery aims at identifying the hub nodes (nodes with a high number of connections). In artificial systems, we propose to consider the group of units belonging to the tail of the considered graph statistics distribution. (E) While in the brain functional role identification can only be achieved by the identification of nodal role in the connectivity network, in ANNs we can perform a network surgery by turning off specific sets of units to evaluate the change in the system performance. This allows associating the network nodal role with a functional role.

not require an oracle on the data or neurogenesis to learn new tasks. In order to study this approach, we considered two simple ANNs architectures (an input layer, a hidden layer, and an output layer) sequentially trained in a simple *handwritten digit recognition task* and in a more complex *face emotion recognition task* using respectively the MNIST database (Deng [2012], LeCun et al. [1989]) and the FER+ database (Goodfellow et al. [2013a], Barsoum et al. [2016]).

In both cases, we train the ANN architectures using different learning strategies (Lomonaco et al. [2021]) in various

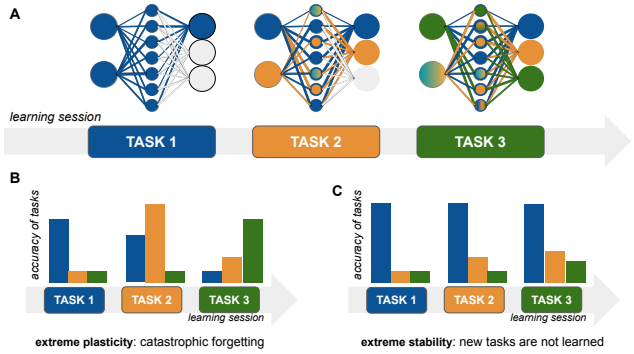


Figure 2: Schematic visualization of the sequential learning framework of three tasks. (**A**) The same ANN is trained sequentially to perform three different tasks. While the architecture does not change across the learning sessions, the associated weights are updated at the end of each session. (**B**) Example of performances of a model affected by catastrophic forgetting. The Task 1 performance, at the end of the third learning session, has deteriorated. This corresponds to ANN where plasticity property is stronger than its stability. (**C**) Example of performance of a model which perfectly learns Task 1, at the end of all the learning sessions. However, the model does not learn new tasks. This corresponds to an ANN where the stability property is stronger than the plasticity. (Adapted from Figure 2 in Hadsell et al. [2020a])

orders (See Section A for details). In particular, we compare brain-inspired learning strategies specifically developed to reduce catastrophic forgetting occurrence (McClelland et al. [1995], French [1999]). The main learning strategies can be grouped into replay methods, regularization methods, and neurogenesis methods (van de Ven and Tolias [2019], Hadsell et al. [2020a], Parisi et al. [2019], Buzzega et al. [2020], Aimone et al. [2009], Draelos et al. [2017]). In this first exploration study, we focus on the analysis of changes happening over a fixed graph structure, thus we do not consider neurogenesis strategies which require a change in the architecture structure as new tasks are learned.

The first family has been inspired by the replay mechanism observed in rodents and human brains (McClelland et al. [1995], O'Reilly et al. [2014], Liu et al. [2019]), where neural activity patterns are replayed during sleep as a means of memory consolidation. For ANNs, two implementations have been introduced: the rehearsal approach where some of the previously seen samples are reused with the current samples to learn, and the pseudo-rehearsal approach where artificially generated new examples are introduced to represent previously learned knowledge. We consider two rehearsal approaches: **Sample Replay** that stores randomly previously seen samples (Lomonaco et al. [2021]) and **GDumb** that selects the stored samples by asymptotically balancing the class distribution (Prabhu et al. [2020]). For the pseudo-rehearsal approach, instead, we evaluate the **Dream Net** strategy (Mainsant et al. [2021]).

The regularization methods have been developed to retain the most important weights while learning new classes. Their bio-inspiration relies on the hypothesis that continual learning relies on task-specific synaptic consolidation, making certain synapses less plastic and stable over time (Clopath [2012]). For instance, experiments with mice demonstrate that a strengthening of excitatory synapses occurs at new skill acquisition (Yang et al. [2009]), leading to an increased volume of specific spines. The increased volume persists despite the subsequent learning of new tasks and is associated with the persistence of performance of the basal task several months later. When these spines are removed the task is forgotten (Cichon and Gan [2015], Hayashi-Takagi et al. [2015]). Among the ANNs regularization methods, we consider Elastic-Weight-Consolidation (**EWC**), Synaptic Intelligence (**SI**) and Learning without Forgetting (**LwF**) strategies (Kirkpatrick et al. [2017], Zenke et al. [2017], Li and Hoiem [2017]). EWC and SI are *structural* regularization methods (Li and Hoiem [2017]) that constrain relevant weights to stay close to their old values. The major difference between these two approaches is given by the estimation of important weights. EWC relies on an offline estimation of the Fisher information matrix while SI proposes an online computation of the importance of a synapse being proportional to the product of its weight and the activity of the post-synaptic neurons. The LwF strategy is a *functional* regularization approach, meaning it penalizes changes in the map input-output of the neural network, by constraining the previous network predictions and the current one to be similar when applied to the new task. More details on the considered learning strategies can be found in Section A and in the referenced papers.

To achieve a baseline comparison, we will also consider **Finetune** and **Cumulative** strategies. In the former nothing is done to avoid catastrophic forgetting, while in the latter the architecture is subsequently trained using all previously seen training data up to the task of the current session as it happens in Offline training.

Since the occurrence of catastrophic forgetting is related to the stability-plasticity dilemma (Abraham and Robins [2005], Mermillod et al. [2013]), we propose to evaluate each architecture and strategy on these two properties by the estimation of the following metrics based on the accuracy, i.e. the proportion of correct predictions divided by the total number of predictions. These metrics have been proposed in Kemker et al. [2018]:

$$stability := \Omega_{\mathbf{base}} = \frac{1}{T-1}\sum_{t=2}^{T}\frac{\alpha_{\text{base,t}}}{\alpha_{\text{ideal}}} \quad (1)$$

$$plasticity := \Omega_{\mathbf{new}} = \frac{1}{T-1}\sum_{t=2}^{T}\alpha_{\text{new,t}} \quad (2)$$

where T corresponds to the total number of learning sessions, $\alpha_{\text{new,t}}$ is the test accuracy for the class immediately learned at session t, $\alpha_{\text{base,t}}$ is the test accuracy of the class learned during the first session (base set) after t new learning sessions and α_{ideal} is the offline method accuracy on the base set, which can be assumed to be the ideal performance. In continual learning, we notate local accuracy the accuracy of the last learned class, and global accuracy at each learning session as the accuracy of all the seen classes.

We quantify the *stability* with $\Omega_{\mathbf{base}}$ that measures the ability to retain the class learned during the first session (session 0), after learning the successive sessions; and the *plasticity* with $\Omega_{\mathbf{new}}$ that measures the performance in learning a new task (See B-C panels in Fig. 2). Unless a model outperforms the offline model accuracy on the base set α_{ideal}, *stability* and *plasticity* vary between $[0, 1]$. Note that the occurrence of the catastrophic forgetting phenomenon is directly quantified by the *stability* metric: low *stability* implies forgetting. From a trained ANN, its corresponding graph model is constructed based on its architecture: the units (i.e nodes) present in each layer and the presence and orientation of edges among the units (as described in Section A). Indeed, we extract the *activation network at rest*, similar to a brain resting-state connectivity analysis (van den Heuvel and Hulshoff Pol [2010]), by feeding to a trained ANN an input sample of 1-entries. Then, we perform a graph filtering procedure to determine the most active units and the strongest connections (**B** panel Fig. 1). This results in an induced graph that is oriented from the input layer to the output layer, and its structure is encoded by its adjacency matrix $A = (a_{lm})$.

Unlike biological connectivity graphs, the feed-forward ANN is a partite graph model with a rigid structure where edges exist only between consecutive layers. This suggests that many of the graph-theoretic measures used in those instances are not amenable to be used in this case. Thus, we focus on characterizing the graph by in and out-degree. We define the statistics $\mathcal{S}_t$ in Eq.3 at each learning session t, corresponding to the difference of in-degree $\deg^{in}$ and out-degree $\deg^{out}$ of each unit i in the hidden layer (**B** panel Fig.1). Specifically, $\deg_t^{out}(i)$ is the number of nodes in the output layer that are adjacent to node i and $\deg_t^{in}(i)$ is the number of nodes in the input layer that are adjacent with node i, counting respectively the number of outgoing and incoming edges in i after t-th training session. Thus,

$$\mathcal{S}_t(i) = \deg_t^{out}(i) - \deg_t^{in}(i). \tag{3}$$

In the following, we refer to an architecture, a learning strategy, a fixed order of training sessions, and the general classification task associated with the database as a **configuration**. For each configuration, we extract as many induced resting-state graphs as the learning sessions. Thus, we track the differences in the statistics across the learning sessions.

At a global level, we show how this brain-inspired framework can be used to extract interpretable statistics which allow the detection of configurations affected by catastrophic forgetting. Moreover, the same statistics can be used to group together configurations that apply the same learning strategies.

Similar to what happens in the human brain, where each node can be associated with a different role given a graph-nodal-statistics (Carboni et al. [2023]), we assume that each unit behaves differently at fostering or inhibiting the plasticity and stability performance of the ANN. We hypothesize that such properties are unevenly distributed across the units of an ANN and we aim at identifying units that contribute differently to the stability and plasticity performances of the ANN. Specifically, we distinguish two subsets of $\mathcal{S}_t$: units that belong to the middle quartiles and units which belong to the tail. As it happens in natural intelligence, where high volume spines persistence is associated with the memory of the corresponding task (Yang et al. [2009]), we hypothesize that units that persist in having extreme $\mathcal{S}$ values can be associated with high stability. In contrast, nodes whose $\mathcal{S}$ values strongly change can be associated with the plasticity property.

In this line, we first propose a general learning strategy characterization **Experiment 1 (Characterization)**. For each configuration, we determine the number of hidden units in the tail of $\mathcal{S}_t$ together with the maximum value of absolute change of $\mathcal{S}_t$ evaluated for the same unit in two consecutive steps as the Maximum Unit Change:

$$\text{Maximum Unit Change} = \max_{\substack{i \in \text{hidden layer,} \\ t \geq 1}} \left| \mathcal{S}_t(i) - \mathcal{S}_{t-1}(i) \right| \tag{4}$$

Next, we define the Consecutive Tail Persistence as the percentage of tail units which appear in the tails of consecutive learning sessions (i.e. in the tail of both $\mathcal{S}_{t-1}$ and $\mathcal{S}_t$). Finally, we define the Tail Persistence as the percentage of units which are in the tail of $\mathcal{S}_1$ after the learning session of the *base* task and in the final distribution $\mathcal{S}_T$ after all learning sessions. We characterize the different learning strategies by comparing these graph-based features and by using them for their identification in a reduced space. Moreover, we validate the use of the degree statistics by investigating the relationship among units in the tail and the norm changes of their synaptic weights across the different learning sessions.

In the **Experiment 2 (Network Surgery)**, we simulate the removal of the increased-volume spines (Cichon and Gan [2015], Hayashi-Takagi et al. [2015]) by pruning (i.e. turning off) the units in ANN. The objective is to associate the interquartile units and the tail units with the stability or plasticity properties of the ANN configuration by evaluating the change in the performance with respect to the standard model (when no unit is turned off). In artificial systems, it is indeed possible to perform such a *network surgery* in order to associate a functional role to a unit following a graph-statistics-based nodal role.

Inspired by the proposal in Zhang et al. [2022], we assume that if a set of units is pruned and the performance changes significantly in terms of *stability*, then the pruned units are critical for the *stability* of the ANN model. The same applies to the *plasticity*. Given the number of units in an ANN, testing the pruning of each unit becomes rapidly unfeasible, thus we consider the set of units in the interquartile and in the tail distribution of $\mathcal{S}$. Hence we propose two pruned ANN versions, one which only preserves the synaptic weights of units in the tail of the distribution of $\mathcal{S}$, and another version that nullifies the weights of the tail units. Since the distribution of $\mathcal{S}_t$ changes across the learning session, we define a copy of each trained ANN in the order sequence, with each pruned copy at the t-th learning session obtained by looking at the $\mathcal{S}_t$ distribution.

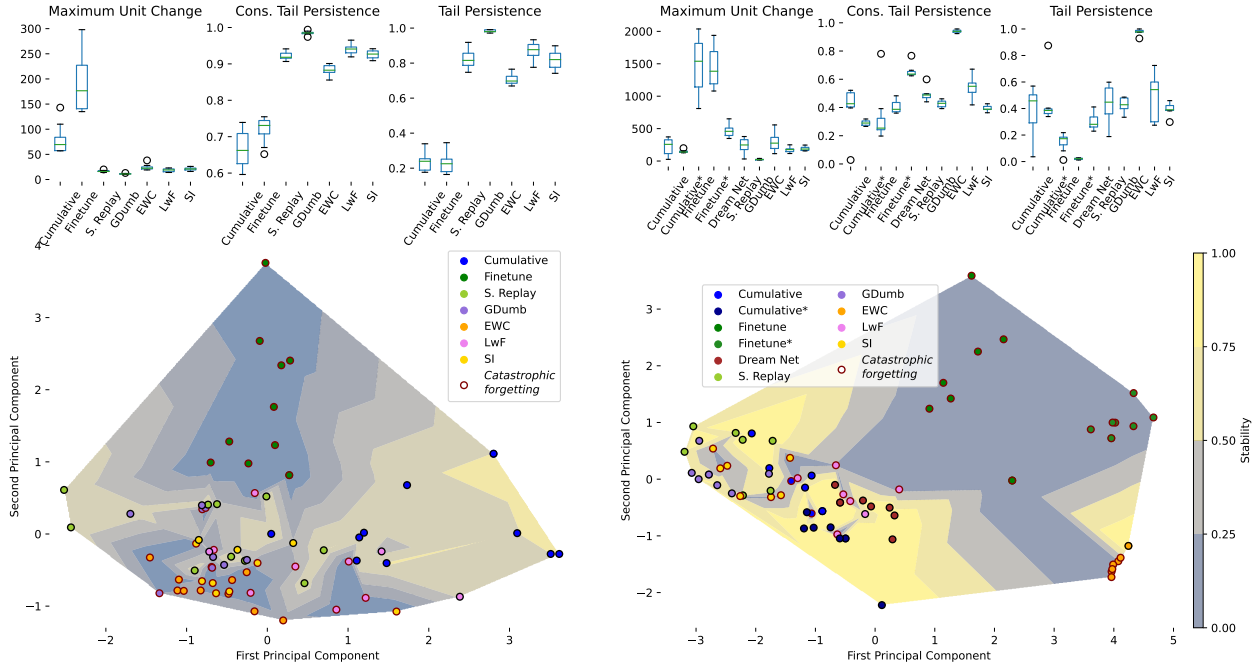


Figure 3: Left: Handwritten Digit Recognition task **Right**: Face Emotion Recognition task. **Top**: Distribution of the graph-based features by strategy. Cons. Tail Persistence: Consecutive Tail Persistence percentage of units in the tail at consecutive learning sessions, Tail Persistence: Percentage of units being in the tail at the first and last learning session. **Bottom**: Visualization of the reduced space of the graph-based features extracted by each sequence of ANNs trained with different strategies. A point corresponds to a unique configuration given by the ANN architecture, the learning strategy, and the fixed-order learning. Points are colored by their stability performances. Each red circle indicates catastrophic forgetting. Similarly, the plasticity performances for all strategies are reported in the appendix. S.Replay: Sample Replay, EWC: Elastic-Weight-Consolidation, SI: Synaptic Intelligence, LwF: Learning without Forgetting

These pruned ANNs ordered by learning sessions are respectively denoted $\mathbb{NN}^{\tau}$ and $\mathbb{NN}^{\bar{\tau}}$, while the standard sequence (i.e. where no pruning is applied) is notated $\mathbb{NN}$.

We evaluate the difference between the *stability/plasticity* of the standard model and its corresponding pruned version: a strong difference corresponds to a critical stability/plasticity unit (i.e. when the units are pruned the performance decreases dramatically), while a low positive difference corresponds to a robust unit (i.e. the pruned units do not affect the performance). Finally, a negative difference identifies plasticity/stability inhibitory units, i.e. units whose synaptic weights have a negative effect on its plasticity/stability.

All our graph modeling analysis of ANNs across sequential learning sessions aims to extract information on how the network adapts to new tasks and how it preserves knowledge of previous tasks from a connectivity point of view. This can potentially provide insights into the plausible neural mechanisms underlying continual learning in the biological neural networks and in reverse inspire the design of more efficient and biologically-plausible continual learning artificial systems.

2. Results

We report the results of each experiment separately.

2.1. Experiment 1: Characterization

2.1.1. Handwritten Digit Recognition task

We observe similar distribution across the different learning strategies, except for the Finetune model. The Maximum Unit Change value is under 100 in continual learning strategies and Cumulative model, but equals 179.7 ± 34.0 for the Finetune model (Fig. 3 Top Left). The tail persistence units shows the peculiar behavior of Cumulative and Finetune models for which the information is subsequently overwritten: their consecutive persistence is severely lower compared to other strategies and only less than 40% of the tail units persists from the learning of the basal task. For all other strategies, the percentage reaches 80% with the exceptional case of GDumb which has an average of 98%.

Visualization in a reduced space of the graph-features vector reveals how these graph-based extracted features are able to identify how Finetune and other strategies are affected by catastrophic forgetting (Fig. 3 Bottom Left). A simple clustering algorithm in the reduced space groups together all regularization learning strategies, replays and Cumulative reaching an overall consensus score of 0.71 (see Section D).

In Fig. 4 we explore the relationship between the nodal statistics and the consecutive norm change of weights in the trained ANN. It is interesting to observe a slight change in norm across consecutive steps even in non-regularization

methods. Additionally, we should notice that in the MNIST dataset, there is a large number of units that receive as input zeros from the border of the flattened images, these units do not update their synaptic weights across the learning process. The statistics $\mathcal{S}$ is able to retrieve the majority of units whose synaptic weights norm does not change across consecutive steps. This is highly valuable for a resting-state analysis which defines induced graphs without any input data.

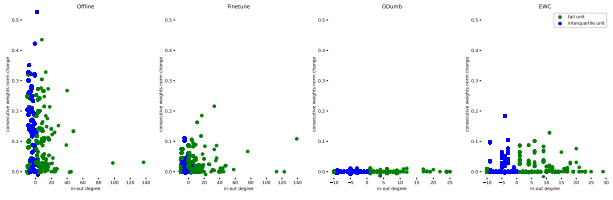


Figure 4: Visualization of the relation between $\mathcal{S}$ and the norm changes of the weights

2.1.2. Face Emotion Recognition task

The same experiment was performed in the Face Emotion Recognition task. In this more difficult task, we considered two different architectures. Since Dream Net requires an auto-hetero associative architecture, we introduced two different versions for Cumulative and Finetune with or without an auto-associative part in the output of the NN. In general, the introduction of auto-associative neurons gives better results already in the standard Cumulative configuration.

As in the handwritten digit recognition task, we report a strong Maximum Unit Change for the Finetune strategy (Top Right Fig. 3). The tail persistence results show a consecutive tail persistence in Dream Net strategy of 0.79 on average, with lower values for Cumulative, Finetune, Sample Replay and GDumb. EWC shows the highest consecutive tail persistence, but the other regularization strategies have on average less persistence with respect Dream Net. Not surprisingly, the percentages in tail persistence have the lowest average for the Finetune configurations and the highest for EWC. Finally, the visualization in a reduced space of the graph features reveals the possibility of identifying the robust learning strategies (Bottom Right Fig.3), a simple K-Means algorithm reaches a consensus score of 0.77. Interestingly, the Finetune configuration, which is dramatically affected by catastrophic forgetting, appears to be moved away with respect to the other clusters.

2.2. Experiment 2: Network Surgery

2.2.1. Handwritten Digit Recognition

We observe different behaviors depending on the learning strategy (see Fig. 5 and additional results in Fig. 16). The Cumulative learning strategy reveals how in offline learning the stability and plasticity properties are shared in the hidden units: pruning of both sets equally affects the performance. At the contrary, the Finetune model, whose stability is extremely low in the standard version, shows strongly improved performance for the pruned sets. Indeed, we can identify the tail units as stability-inhibitory: pruning them has a beneficial effect in recalling the basal task, reaching a maximum *stability* performance of almost 0.46 for the first training session on the class corresponding to the digit 4 (see additional Fig. 16). The GDumb approach behaves similarly to the Finetune or the Cumulative models depending on the standard performance: it exhibits a beneficial effect of pruning the tail when the standard performance is poor and critical behavior in general pruning when the standard performance is higher. A stronger effect is observed when pruning the interquartile range units.

Concerning plasticity, a common general behavior across all learning strategies is a stronger plasticity-critical behavior of the tail units: when their weights are set to zero the plasticity performance is dramatically reduced. The pruning of the interquartile range also affects the plasticity, but with less effect, especially for the Finetune model. We report in details the local accuracy at each learning session (Right 5). We can notice how in the Finetune model, the $\mathbb{NN}^{\tau}$ has positive local accuracy and for a few learning sessions has comparable results with the standard version. In the regularization methods, a transition between the best local accuracy pruned performance exists, with the interquartile having better results at the beginning of the learning sessions and the tail in the following phases.

2.2.2. Face Emotion Recognition task

The Cumulative model distributes the stability properties across all units independently by the distribution of $\mathcal{S}$. This leads to a dramatic decreases of the stability performance for the pruning versions. Concerning the plasticity property, we detect a higher criticality for the units in the interquartile range. For Dream Net, all pruned settings have very poor results for new tasks, with the extreme case of the model with tailing removing $\mathbb{NN}^{\bar{\tau}}$ that has a zero Ω_{new}. Different results are observed for the Finetune model, where the pruning versions, keeping only the weights in the tail, reach the same Ω_{new} that the standard one. Unlike the Cumulative and Dream Net models where none pruned copy reaches similar performances, for Finetune, the $\mathbb{NN}^{\tau}$ is a good pruned copy of the complete standard model.

3. Discussion

In this study, we leverage a novel research framework in which ANNs are studied starting from their connectivity properties. This framework enable us to integrate the biological inspiration of ANNs into their analyzing tool by proposing a way to fill the gap between brain connectivity studies and the analysis of the information flow in ANN.

Utilizing this research framework, we concentrate our analysis on the catastrophic forgetting issue. Our objective was to determine the relationship between existing learning strategies that alleviate the catastrophic forgetting phenomenon and general graph connectivity features. We showed that a simple graph-induced definition and the extraction of interpretable graph features are important indicators of the stability properties of an ANN model which

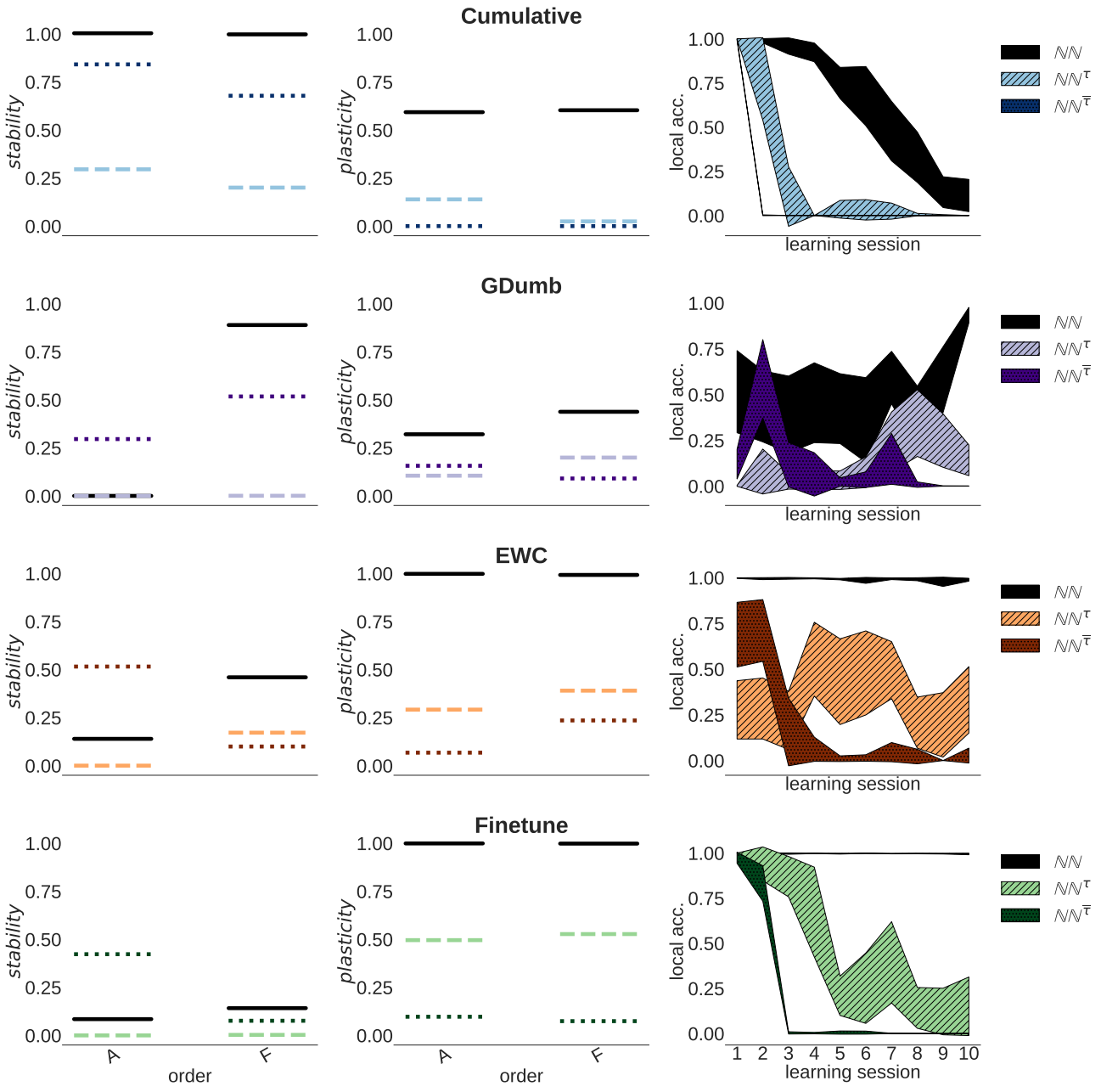


Figure 5: *Stability* (**Left**) and *Plasticity* (**Center**) performances for all configurations on handwritten digit recognition task for the standard model $\mathbb{NN}$ (plain black line) and their pruned versions $\mathbb{NN}^{\tau}, \mathbb{NN}^{\bar{\tau}}$ (colored dashed lines). The performance is reported for two different orders (A,F, see Tab. 2) of learning sessions. **Right**: average local accuracy performances for the last learned class across learning sessions.

are enough to detect the learning strategies applied to the same ANN architecture learning to perform the same task.

This study investigates the utility of implementing memory consolidation or task-specific synaptic consolidation, by distributing unequally the stability and plasticity properties in the ANN units. We report good stability in online learning strategies in the presence of a persistence of tail units with strong synaptic weights across the learning sessions. While such persistence can be expected for the regularization methods, it surprisingly appears to be present in replay methods too, suggesting that both methods can induce similar distribution in the strength of the connection in the hidden units. Surprisingly, replay methods achieve the same result of constraining synaptic weight updates across learning sessions, without requiring hyperparameter tuning, unlike regularization strategies.

Our graph modeling and the chosen graph statistics are able to detect the units whose weights slightly change in

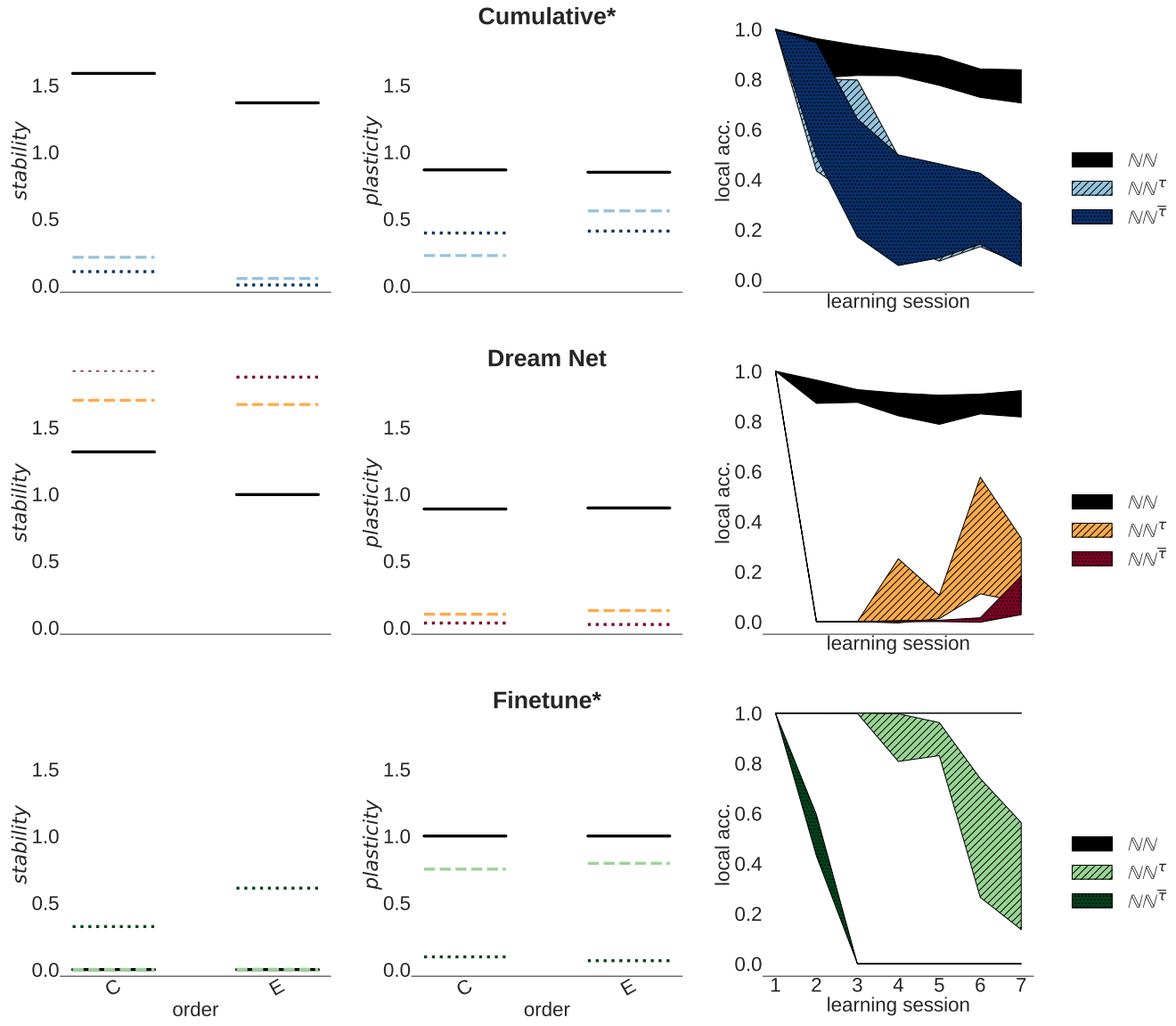


Figure 6: *Stability* (**Left**) and *Plasticity* (**Center**) performances for all strategies on face emotion recognition task for the standard model $\mathbb{NN}$ (plain black line) and their pruned versions $\mathbb{NN}^{\tau}, \mathbb{NN}^{\bar{\tau}}$ (colored dashed lines). The performance is reported for two different orders (A,F, see Tab. 2) of learning sessions. **Right**: average local accuracy performances for the last learned class across learning sessions.

norm across the learning session. This is highly valuable for our graph modeling at rest, which does not estimate the importance of weights given true input data, but whose only relation with data is given by the determined weights after training.

When coming to more complicated tasks, major differences are revealed between the Cumulative and the Dream Net replay methods. First, Dream Net and Cumulative have opposite behavior in the network surgery experiment: any pruning technique of the Cumulative configuration destroys the stability, while for Dream Net the pruning increases stability at the cost of diminishing plasticity. This suggests that the use of a dropout technique in the training phase of Dream Net, can result in better stability results.

The results in the pruning experiments capture the difference between multiple tasks learning in one session and sequential learning: while offline learning automatically distributes the connections and their strength across the hidden units for the different tasks, sequential learning imposes strong task-specific synaptic weights on a few connections which are continuously overwritten and substituted. Thus, catastrophic forgetting is alleviated when these connections are not erased but slightly adapted to the subsequent tasks.

In the Finetune model, the proposed graph statistics detect the lottery ticket winner (i.e. a pruned version of the model with the same performance as the entire network and less redundancy (Frankle and Carbin [2018])). We found that the model which only preserves the weights of the tail units reaches the same performance of the standard one, despite

the number of preserved units. This is in line with pruning literature results which determine a very small subnetwork having almost the same performance as the complete model Wolinski [2020], Tanaka et al. [2020]. Remarkably, we prove that the high plasticity of a Finetune model is strongly related by units in the tail of the distribution and that a simple weights-injection across sequential models can enhance its stability.

Besides laying down the foundation for a graph-statistics-based study of the learning process in ANN, we mainly show empirical findings of post-training ANN model characterization. However, the training process in an ANN is not negligible and by definition highly dependent on the dataset used for training (Ramyachitra and Manikandan [2014], Ali et al. [2019], Djolonga et al. [2021], Song et al. [2022]). This is the major weakness preventing results generalization of a post-training analysis: results may change when different tasks or datasets are used, or even by testing a different learning order. We tackle some generalization induced by the order of learning sessions, by randomly testing different orders since testing all possible orders becomes quickly intractable.

In addition, our proposal is only applied to a feed-forward ANN model having a unique hidden layer. In deeper neural networks, we may observe different results depending on the considered hidden layers, as different robustness was observed associated with different layers (Zhang et al. [2022]).

Many studies have already introduced the need for more complex neural network architectures, giving both brain-inspired motivations and graph-based topological requirements (Sussillo and Abbott [2009], Mocanu et al. [2016, 2018], Hasson et al. [2020], Liu et al. [2021], Kaviani and Sohn [2021]), hence indicating possible future research directions to extend the present work to different architectures.

In addition to existing work that promotes the graph-based approach to define new ANN architectures (Elsken et al. [2019], Leijnen and Veen [2020]) we introduce for the first time, the graph-based analysis of the connectivity at rest of ANNs. Related works are presented in Section C.

We have used graph modeling and graph-based statistics to analyze trained ANNs in a continual learning framework. By studying different models which differently address the catastrophic forgetting issue, we show that it is possible to identify models with different performances based on simple features extracted from a binary graph obtained considering the strongest weights connection. We propose to identify critical hidden units according to the performance of pruned version model which turns off neurons according to their in-out degree values in the induced graph. The results show that the selected statistics and choice of hidden unit sets in the tail or interquartile range can be used to identify critical unit sets for the stability and plasticity of the corresponding configuration.

Finally, as it happens in the brain, we conclude that the hidden units in ANN are not homogeneous in recalling previously learned information or at adapting to newly learned information. These results lay down the foundations to study how the learning process in ANN using graph-theory tools, providing insights into the occurrence of catastrophic forgetting and the presence of a stability-critical set of neurons.

4. Acknowledgments

L. Carboni and D. Nwaigwe are the recipients of a grant from MIAI@Grenoble Alpes (ANR 19-P3IA-003).

References

Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

Keshab K Parhi and Nanda K Unnikrishnan. Brain-inspired computing: Models and architectures. *IEEE Open Journal of Circuits and Systems*, 1:185–204, 2020.

Matthew Botvinick, Jane X Wang, Will Dabney, Kevin J Miller, and Zeb Kurth-Nelson. Deep reinforcement learning and its neuroscientific implications. *Neuron*, 107(4):603–616, 2020.

Demis Hassabis, Dharshan Kumaran, Christopher Summerfield, and Matthew Botvinick. Neuroscience-inspired artificial intelligence. *Neuron*, 95(2):245–258, 2017.

Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5:115–133, 1943.

JL McClelland and DE Rumelhart. Parallel distributed processing: Explorations in the microstructureofcognition (vo\. 2), 1986.

Donald Olding Hebb. *The organization of behavior: A neuropsychological theory*. Psychology press, 2005.

Adam H. Marblestone, Greg Wayne, and Konrad P. Kording. Toward an integration of deep learning and neuroscience. *Frontiers in Computational Neuroscience*, 10:94, 2016. ISSN 1662-5188. doi: 10.3389/fncom.2016.00094.

Lyes Khacef, Nassim Abderrahmane, and Benoît Miramond. Confronting machine-learning with neuroscience for neuromorphic architectures design. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2018.

Wolfgang Maass. Networks of spiking neurons: the third generation of neural network models. *Neural networks*, 10(9):1659–1671, 1997.

Daniel LK Yamins, Ha Hong, Charles F Cadieu, Ethan A Solomon, Darren Seibert, and James J DiCarlo. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the national academy of sciences*, 111(23):8619–8624, 2014.

Radoslaw Martin Cichy, Aditya Khosla, Dimitrios Pantazis, Antonio Torralba, and Aude Oliva. Comparison of deep neural networks to spatio-temporal cortical dynamics of human visual object recognition reveals hierarchical correspondence. *Scientific reports*, 6(1):1–13, 2016.

D. L. Yamins and J. J. DiCarlo. Using goal-driven deep learning models to understand sensory cortex. *Nat Neurosci*, 19(3):356–65, 2016. ISSN 1097-6256. doi: 10.1038/nn.4244.

Ilya Kuzovkin, Raul Vicente, Mathilde Petton, Jean-Philippe Lachaux, Monica Baciu, Philippe Kahane, Sylvain Rheims, Juan R Vidal, and Jaan Aru. Activations of deep convolutional neural networks are aligned with gamma band activity of human visual cortex. *Communications biology*, 1(1):1–12, 2018.

P. Bashivan, K. Kar, and J. J. DiCarlo. Neural population control via deep image synthesis. *Science*, 364(6439), 2019. doi: 10.1126/science.aav9436.

Steve Petersen and Olaf Sporns. Brain networks and cognitive architectures. *Neuron*, 88:207–219, 10 2015. doi: 10.1016/j.neuron.2015.09.027.

Ed Bullmore and Olaf Sporns. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature reviews neuroscience*, 10(3):186–198, 2009.

Jinhui Wang, Xinian Zuo, and Yong He. Graph-based network analysis of resting-state functional mri. *Frontiers in systems neuroscience*, page 16, 2010.

Olaf Sporns. Graph theory methods: applications in brain networks. *Dialogues in clinical neuroscience*, 2022.

Albert-László Barabási. Network science. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1987):20120375, 2013.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *Advances in neural information processing systems*, 33:15920–15930, 2020.

Raia Hadsell, Dushyant Rao, Andrei A. Rusu, and Razvan Pascanu. Embracing change: Continual learning in deep neural networks. *Trends in Cognitive Sciences*, 24(12):1028–1040, 2020a. ISSN 1364-6613. doi: https://doi.org/10.1016/j.tics.2020.09.004.

Raia Hadsell, Dushyant Rao, Andrei A Rusu, and Razvan Pascanu. Embracing change: Continual learning in deep neural networks. *Trends in cognitive sciences*, 24(12):1028–1040, 2020b.

Norton W Milgram, Colin M MacLeod, and Ted L Petit. *Neuroplasticity, learning, and memory.* Alan R. Liss, 1987.

Alvaro Pascual-Leone, Amir Amedi, Felipe Fregni, and Lotfi B Merabet. The plastic human brain cortex. *Annu. Rev. Neurosci.*, 28:377–401, 2005.

Kaitlyn Casimo. *Spontaneous and task-related changes in resting state connectivity*. PhD thesis, University of Washington, 2018.

Fabrizio de Vico Fallani, Farhan Baluch, Laura Astolfi, Devika Subramanian, George Zouridakis, and Fabio Babiloni. Structural organization of functional networks from eeg signals during motor learning tasks. *International Journal of Bifurcation and Chaos*, 20(03):905–912, 2010.

George Zouridakis, Farhan Baluch, Ian Stevenson, and Devika Subramanian. Spatiotemporal profiles of brain activation during learning and strategy formulation. In *Joint Meeting of the 6th International Symposium on Noninvasive Functional Source Imaging of the Brain and Heart and the International Conference on Functional Biomedical Imaging*, pages 323–326, 2007. doi: 10.1109/NFSI-ICFBI.2007.4387765.

Marion Mainsant, Miguel Solinas, Marina Reyboz, Christelle Godin, and Martial Mermillod. Dream net: a privacy preserving continual leaming model for face emotion recognition. In *2021 9th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*, pages 01–08. IEEE, 2021.

Miguel Solinas, Stéphane Rousset, Romain Cohendet, Yannick Bourrier, Marion Mainsant, A Molnos, Marina Reyboz, and Martial Mermillod. Beneficial effect of combined replay for continual learning. In *ICAART (2)*, pages 205–217, 2021.

Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

Ian J Goodfellow, Dumitru Erhan, Pierre Luc Carrier, Aaron Courville, Mehdi Mirza, Ben Hamner, Will Cukierski, Yichuan Tang, David Thaler, Dong-Hyun Lee, et al. Challenges in representation learning: A report on three machine learning contests. In *International conference on neural information processing*, pages 117–124. Springer, 2013a.

Emad Barsoum, Cha Zhang, Cristian Canton Ferrer, and Zhengyou Zhang. Training deep networks for facial expression recognition with crowd-sourced label distribution. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, pages 279–283, 2016.

Vincenzo Lomonaco, Lorenzo Pellegrini, Andrea Cossu, Antonio Carta, Gabriele Graffieti, Tyler L Hayes, Matthias De Lange, Marc Masana, Jary Pomponi, Gido M Van de Ven, et al. Avalanche: an end-to-end library for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3600–3610, 2021.

James L McClelland, Bruce L McNaughton, and Randall C O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review*, 102(3):419, 1995.

Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999.

Gido M. van de Ven and Andreas S. Tolias. Three scenarios for continual learning, 2019. URL https://arxiv.org/abs/1904.07734.

German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019.

James B. Aimone, Janet Wiles, and Fred H. Gage. Computational influence of adult neurogenesis on memory encoding. *Neuron*, 61(2):187–202, 2009.

Timothy J Draelos, Nadine E Miner, Christopher C Lamb, Jonathan A Cox, Craig M Vineyard, Kristofor D Carlson, William M Severa, Conrad D James, and James B Aimone. Neurogenesis deep learning: Extending deep networks to accommodate new classes. In *2017 international joint conference on neural networks (IJCNN)*, pages 526–533. IEEE, 2017.

Randall C O'Reilly, Rajan Bhattacharyya, Michael D Howard, and Nicholas Ketz. Complementary learning systems. *Cognitive science*, 38(6):1229–1248, 2014.

Yunzhe Liu, Raymond J Dolan, Zeb Kurth-Nelson, and Timothy EJ Behrens. Human replay spontaneously reorganizes experience. *Cell*, 178(3):640–652, 2019.

Ameya Prabhu, Philip HS Torr, and Puneet K Dokania. Gdumb: A simple approach that questions our progress in continual learning. In *European conference on computer vision*, pages 524–540. Springer, 2020.

Claudia Clopath. Synaptic consolidation: an approach to long-term learning. *Cognitive neurodynamics*, 6(3):251–257, 2012.

Guang Yang, Feng Pan, and Wen-Biao Gan. Stably maintained dendritic spines are associated with lifelong memories. *Nature*, 462(7275):920–924, 2009.

Joseph Cichon and Wen-Biao Gan. Branch-specific dendritic ca2+ spikes cause persistent synaptic plasticity. *Nature*, 520(7546):180–185, 2015.

Akiko Hayashi-Takagi, Sho Yagishita, Mayumi Nakamura, Fukutoshi Shirai, Yi I Wu, Amanda L Loshbaugh, Brian Kuhlman, Klaus M Hahn, and Haruo Kasai. Labelling and optical erasure of synaptic memory traces in the motor cortex. *Nature*, 525(7569):333–338, 2015.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, pages 3987–3995. PMLR, 2017.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

Wickliffe C. Abraham and Anthony Robins. Memory retention – the synaptic stability versus plasticity dilemma. *Trends in Neurosciences*, 28(2):73–78, 2005.

Martial Mermillod, Aurélia Bugaiska, and Patrick Bonin. The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects, 2013.

Ronald Kemker, Marc McClure, Angelina Abitino, Tyler Hayes, and Christopher Kanan. Measuring catastrophic forgetting in neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Martijn P. van den Heuvel and Hilleke E. Hulshoff Pol. Exploring the brain network: A review on resting-state fmri functional connectivity. *European Neuropsychopharmacology*, 20(8):519–534, 2010.

Lucrezia Carboni, M Dojat, and S Achard. Nodal statistics-based equivalence relation for graph collections. *Physical Reviews E*, 107:014302–1–14, 2023.

Chiyuan Zhang, Samy Bengio, and Yoram Singer. Are all layers created equal? *Journal of Machine Learning Research*, 23:1–28, 2022.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding small, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

Pierre Wolinski. *Structural Learning of Neural Networks*. PhD thesis, Université Paris-Saclay, 2020.

Hidenori Tanaka, Daniel Kunin, Daniel L Yamins, and Surya Ganguli. Pruning neural networks without any data by iteratively conserving synaptic flow. *Advances in neural information processing systems*, 33: 6377–6389, 2020.

D Ramyachitra and Parasuraman Manikandan. Imbalanced dataset classification and solutions: a review. *International Journal of Computing and Business Research (IJCBR)*, 5(4):1–29, 2014.

Haseeb Ali, Mohd Najib Mohd Salleh, Rohmat Saedudin, Kashif Hussain, and Muhammad Faheem Mushtaq. Imbalance class problems in data mining: A review. *Indonesian Journal of Electrical Engineering and Computer Science*, 14(3):1560–1571, 2019.

Josip Djolonga, Jessica Yung, Michael Tschannen, Rob Romijnders, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Matthias Minderer, Alexander D'Amour, Dan Moldovan, et al. On robustness and transferability of convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16458–16468, 2021.

Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

David Sussillo and Larry F Abbott. Generating coherent patterns of activity from chaotic neural networks. *Neuron*, 63(4):544–557, 2009.

Decebal Constantin Mocanu, Elena Mocanu, Phuong H Nguyen, Madeleine Gibescu, and Antonio Liotta. A topological insight into restricted boltzmann machines. *Machine Learning*, 104:243–270, 2016.

Decebal Constantin Mocanu, Elena Mocanu, Peter Stone, Phuong H Nguyen, Madeleine Gibescu, and Antonio Liotta. Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science. *Nature communications*, 9(1):2383, 2018.

Uri Hasson, Samuel A Nastase, and Ariel Goldstein. Direct fit to nature: An evolutionary perspective on biological and artificial neural networks. *Neuron*, 105(3):416–434, 2020.

Shiwei Liu, Tim Van der Lee, Anil Yaman, Zahra Atashgahi, Davide Ferraro, Ghada Sokar, Mykola Pechenizkiy, and Decebal Constantin Mocanu. Topological insights into sparse neural networks. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part III*, pages 279–294. Springer, 2021.

Sara Kaviani and Insoo Sohn. Application of complex systems topologies in artificial neural networks optimization: An overview. *Expert Systems with Applications*, 180:115073, 2021.

Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Neural architecture search: A survey. *The Journal of Machine Learning Research*, 20(1): 1997–2017, 2019.

Stefan Leijnen and Fjodor van Veen. The neural network zoo. *Multidisciplinary Digital Publishing Institute Proceedings*, 47(1):9, 2020.

Gaurav Jain and Jason Ko. Handwritten digits recognition. *Multimedia Systems, Project Report, University of Toronto*, pages 1–3, 2008.

Raymond Legault, Ching Y. Suen, and Christine Nadal. Difficult cases in handwritten numeral recognition. In Henry S. Baird, Horst Bunke, and Kazuhiko Yamamoto, editors, *Structured Document Image Analysis*, pages 235–249. Springer Berlin Heidelberg, Berlin, Heidelberg, 1992.

Yann LeCun, Larry Jackel, Leon Bottou, A Brunot, Corinna Cortes, John Denker, Harris Drucker, Isabelle Guyon, UA Muller, Eduard Sackinger, et al. Comparison of learning algorithms for handwritten digit recognition. In *International conference on artificial neural networks*, volume 60(1), pages 53–60. Perth, Australia, 1995.

Dan Ciregan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3642–3649. IEEE, 2012.

Narender Kumar and Himanshu Beniwal. Survey on handwritten digit recognition using machine learning. *International Journal of Computer Sciences and Engineering*, 6(05):96–100, 2018.

Xiao-Xiao Niu and Ching Y. Suen. A novel hybrid cnn–svm classifier for recognizing handwritten digits. *Pattern Recognition*, 45(4):1318–1325, 2012.

Samay Pashine, Ritik Dixit, and Rishika Kushwah. Handwritten digit recognition using machine and deep learning algorithms. *CoRR*, abs/2106.12614, 2021. URL https://arxiv.org/abs/2106.12614.

Alejandro Baldominos, Yago Saez, and Pedro Isasi. A survey of handwritten character recognition with mnist and emnist. *Applied Sciences*, 9 (15):3169, 2019.

Kevin Jarrett, Koray Kavukcuoglu, Marc'Aurelio Ranzato, and Yann LeCun. What is the best multi-stage architecture for object recognition? In *2009 IEEE 12th International Conference on Computer Vision*, pages 2146–2153, 2009. doi: 10.1109/ICCV.2009.5459469.

Ian J. Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks, 2013b. URL https://arxiv.org/abs/1312.6211.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Ralph Adolphs, Daniel Tranel, Hanna Damasio, and Antonio Damasio. Impaired recognition of emotion in facial expressions following bilateral damage to the human amygdala. *Nature*, 372(6507):669–672, 1994.

Ralph Adolphs, Daniel Tranel, Hanna Damasio, and Antonio R Damasio. Fear and the human amygdala. *Journal of neuroscience*, 15(9):5879–5891, 1995.

Andrew J Calder. Facial emotion recognition after bilateral amygdala damage: Differentially severe impairment of fear. *Cognitive Neuropsychology*, 13(5):699–745, 1996.

Hans C Breiter, Nancy L Etcoff, Paul J Whalen, William A Kennedy, Scott L Rauch, Randy L Buckner, Monica M Strauss, Steven E Hyman, and Bruce R Rosen. Response and habituation of the human amygdala during visual processing of facial expression. *Neuron*, 17(5):875–887, 1996.

John S Morris, Christopher D Frith, David I Perrett, Daniel Rowland, Andrew W Young, Andrew J Calder, and Raymond J Dolan. A differential neural response in the human amygdala to fearful and happy facial expressions. *Nature*, 383(6603):812–815, 1996.

Michael Davis and Paul J Whalen. The amygdala: vigilance and emotion. *Molecular psychiatry*, 6(1):13–34, 2001.

Luiz Pessoa and Ralph Adolphs. Emotion processing and the amygdala: from a'low road'to'many roads' of evaluating biological significance. *Nature reviews neuroscience*, 11(11):773–782, 2010.

Adam K Anderson and Elizabeth A Phelps. Expression without recognition: contributions of the human amygdala to emotional communication. *Psychological Science*, 11(2):106–111, 2000.

Paolo Fusar-Poli, Anna Placentino, Francesco Carletti, Paola Landi, Paul Allen, Simon Surguladze, Francesco Benedetti, Marta Abbamonte, Roberto Gasparotti, Francesco Barale, et al. Functional atlas of emotional faces processing: a voxel-based meta-analysis of 105 functional magnetic resonance imaging studies. *Journal of Psychiatry and Neuroscience*, 34(6):418–432, 2009.

Alejandro Gómez, O Lucia Quintero, Natalia Lopez-Celani, and Luisa F Villa. Emotional networked maps from eeg signals. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 34–37. IEEE, 2020.

Zhijie Liao, Tobias Banaschewski, Arun LW Bokde, Sylvane Desrivières, Herta Flor, Antoine Grigis, Hugh Garavan, Penny Gowland, Andreas Heinz, Bernd Ittermann, et al. Similarity and stability of face network across populations and throughout adolescence and adulthood. *NeuroImage*, 244:118587, 2021.

Raphael Underwood, Eva Tolmeijer, Johannes Wibroe, Emmanuelle Peters, and Liam Mason. Networks underpinning emotion: A systematic review and synthesis of functional and effective connectivity. *NeuroImage*, 243: 118486, 2021.

Jyoti Kumari, Reghunadhan Rajesh, and KM Pooja. Facial expression recognition: A survey. *Procedia computer science*, 58:486–491, 2015.

Dhwani Mehta, Mohammad Faridul Haque Siddiqui, and Ahmad Y Javaid. Facial emotion recognition: A survey and real-world user experiences in mixed reality. *Sensors*, 18(2):416, 2018.

Shan Li and Weihong Deng. Deep facial expression recognition: A survey. *IEEE transactions on affective computing*, 2020.

Paul Ekman. Are there basic emotions? *Psychological Review*, 99(3), 1992a.

Paul Ekman. Facial expressions of emotion: an old controversy and new findings. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 335(1273):63–69, 1992b.

Paul Ekman and Wallace V Friesen. Facial action coding system. *Environmental Psychology & Nonverbal Behavior*, 1978.

XuMing Wang, Jin Huang, Jia Zhu, Min Yang, and Fen Yang. Facial expression recognition with deep learning. In *Proceedings of the 10th International Conference on Internet Multimedia Computing and Service*, pages 1–4, 2018.

Emanuele La Malfa, Gabriele La Malfa, Giuseppe Nicosia, and Vito Latora. Characterizing learning dynamics of deep neural networks via complex networks. In *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 344–351. IEEE, 2021.

Emanuele La Malfa, Gabriele La Malfa, Claudio Caprioli, Giuseppe Nicosia, and Vito Latora. Deep neural networks as complex networks. *arXiv preprint arXiv:2209.05488*, 2022.

Jonas Richiardi, Sophie Achard, Horst Bunke, and Dimitri Van De Ville. Machine learning with brain graphs: predictive modeling approaches for functional imaging in systems neuroscience. *IEEE Signal processing magazine*, 30(3):58–70, 2013.

Alex Fornito, Andrew Zalesky, and Michael Breakspear. Graph analysis of the human connectome: promise, progress, and pitfalls. *Neuroimage*, 80:426–444, 2013.

Fabrizio De Vico Fallani, Vito Latora, and Mario Chavez. A topological criterion for filtering information in complex brain networks. *PLoS computational biology*, 13(1):e1005305, 2017.

Matteo Zambra, Amos Maritan, and Alberto Testolin. Emergence of network motifs in deep neural networks. *Entropy*, 22(2), 2020. ISSN 1099-4300. doi: 10.3390/e22020204. URL https://www.mdpi.com/1099-4300/22/2/204.

Ali Masoudi-Nejad, Falk Schreiber, and Zahra Razaghi Moghadam Kashani. Building blocks of biological networks: a review on major network motif discovery algorithms. *IET systems biology*, 6(5):164–174, 2012.

Sabyasachi Patra and Anjali Mohapatra. Review of tools and algorithms for network motif discovery in biological networks. *IET systems biology*, 14 (4):171–189, 2020.

Leonardo FS Scabini and Odemir M Bruno. Structure and performance of fully connected neural networks: Emerging complex network properties. *arXiv preprint arXiv:2107.14062*, 2021.

Blaise Hanczar, Farida Zehraoui, Tina Issa, and Mathieu Arles. Biological interpretation of deep neural network for phenotype prediction based on gene expression. *BMC bioinformatics*, 21(1):1–18, 2020.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

Ciprian A Corneanu, Meysam Madadi, Sergio Escalera, and Aleix M Martinez. What does it mean to learn in deep networks? and, how does one detect adversarial attacks? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4757–4766, 2019.

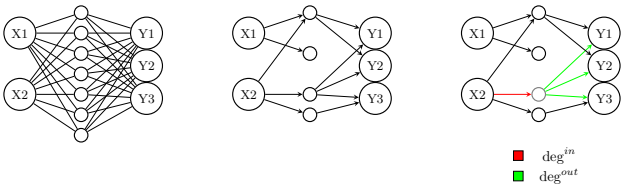


Figure 7: A toy example of an induced graph. Left: NN architecture graph. Center: The induced graph at rest. Right: Visualization of the statistics of interest for the gray node.

A. Material and Method

A.1. Method

A.1.1. Activation Network and Induced Graph Definition

We consider in our study only feedforward ANNs, where edges are oriented from the input layer to the output layer. An artificial neuron is a parametric function $f_{\mathbf{w},b}$ which transforms an input x vector into an output y. Similarly, an artificial neural network (ANN) is a mathematical function that processes information from the input to the output by applying a combination of artificial neurons. The way the neurons or units are combined together determines the architecture of the ANN which can be easily represented as a graph, whose vertices are the units and oriented edges represent a linear transformation applied to the output of the first unit and taken as input by the second unit. The architecture graph determines the number of layers (list of neurons having the same input), the number of units per layer, and the presence and orientation of edges among the units. Here, a unit represents a nonlinear transformation function, defined through an activation function that mimics the stimulation of a biological neuron. Some common examples of activation functions are the sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$ or the Rectified Linear Unit (ReLU) function $\sigma(x) = \max(0, x)$ Fig. 8. In our proposal, we consider an ANN model to be

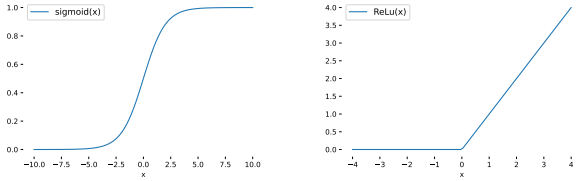


Figure 8: Examples of common activation functions

uniquely identified by an architecture graph and a synaptic weight function which determines the parameters of each artificial neuron by associating a weight to each edge.

The weights are uniquely determined at the end of the training process, we notate $\mathbf{w}_{i,k}^{l-1,l}$ the weight associated with the edge connecting the i-th node of layer $l-1$ and the k-th node of layer l.

At each input sample x processed by the ANN model, we can associate an activation network defined by the computation of the sequence of linear and nonlinear transformations

applied to the input and associate to each unit, its activation value f and to each edge, the resulting value of the corresponding transformation. In biological terms, the activation network can be thought of as the map of the brain response to a stimulus x. In particular, each unit i in layer l is associated with an activation u_i:

$$u_i^l = \sigma\Big(\sum_j \mathbf{w}_{j,i}^{l-1,l} u_j^{l-1} + b_i\Big) \tag{5}$$

where σ is the activation function and b is a fixed bias term, $\mathbf{w}$ and b are the parameters of the artificial neuron function.

Inspired by the concept of brain resting-state connectivity analysis, where the brain activation map is determined at mind-wandering, we define the *activation network at rest* feeding to the ANN a vector of ones to simulate the process of a *non-task-related-information*. With this procedure, the ANN is not engaged in any specific cognitive task, and we can assess the intrinsic functional organization of the artificial system. The activation network at rest results in a set of artificial neurons that are spontaneously active and functionally connected with each other in the absence of true external inputs. The vector of ones can serve as a simple way to activate the network at rest, by providing a constant input to all the nodes, yet different choices (random noise, periodic signals, average of the considered data, etc.) can be envisaged in future explorations. Thus, for the units in the input layer, we fix suggestion: put a tilde on this since it corresponds to a quantity dervied from "activation state". See other comment in blue related to this where i discuss tilde vs non-tilde quantities $u_i^0 = 1$. The weights of the edges in the following layers of the activation network at rest are given by

$$\tilde{w}_{i,k}^{l-1,l} = \mathbf{w}_{i,k}^{l-1,l} u_i^{l-1} \tag{6}$$

Hence, for the first layer, we have $\tilde{w}_{i,k}^{0,1} = \mathbf{w}_{i,k}^{0,1} u_i^0 = \mathbf{w}_{i,k}^{0,1}$ and for the subsequent layer, we simply apply (6) again with u_i^1 the output of the first layer. Even if for the first layer $\tilde{w}_{i,k}^{0,1}$ equals $\mathbf{w}_{i,k}^{0,1}$, they formally correspond to two very distinct concepts, $\mathbf{w}$ being the parametric synaptic weights of the artificial neural network function and $\tilde{w}$ the weights function associated to the edges of a graph.

Finally, we perform a graph filtering procedure for both the nodal features - to determine the most active units - and edges weights - to determine the strongest connections. As final results of this procedure, we obtain a binary graph, i.e. every two nodes are either connected either disconnected. Note that the number of nodes and edges in the activation network only depends on the architecture graph of the model which is fixed for all configurations. The graph filtering procedure can be performed by choosing the number of units to extract in each layer, and the desired total number of edges with respect to a given criterion (for instance to observe particular graph properties). This corresponds to determining a weight threshold WT such that the number of edges whose graph weights are greater in absolute value of WT equals the chosen graph sparsity. Similarly, on the nodal features, we can filter out units whose activation is not greater of an activation threshold AT. We notate the induced graph obtained by filtering the activation network at rest of a trained neural network as $\mathcal{G}$(NN)

suggestion: clearly distinguish between quantities with tilde and those without. It makes sense to set non-tilde quantities equal to generic activations/ weights of the neural network, while letting tilde values represent the neural network quantities that result when starting with the activation of ones

how about this for binary graph definition?:

$$\mathcal{G}(\mathrm{NN}) = (\mathcal{V}, \mathcal{E}) : \tag{7}$$

$$\mathcal{V} = \{\tilde{u}_i^l : \tilde{u}_i^l > AT\} \tag{8}$$

$$\mathcal{E} = \{(\tilde{u}_i^{l-1}, \tilde{u}_j^l)\} : |\tilde{w}_{i,j}^{l,l-1}| > \mathrm{WT} \text{ and } \tilde{u}_i^{l-1}, \tilde{u}_j^l \text{ are in } \mathcal{V}\} \tag{9}$$

$$\mathcal{G}(\mathrm{NN}) = (\mathcal{V}, \mathcal{E}) : \tag{10}$$

$$\mathcal{V} = \{u^0(i), \forall i \in 1, \dots, H(0)\} \cup \{(l-1, i), \forall l \in \{2, \dots, D\}, i \in \{1, \dots, H(l)\}, u^{l-1}(i) > AT\} \tag{11}$$

$$A = (a_{(l-1,i),(l,j)})_{(l-1,i),(l,j)\in\mathcal{V}} \tag{12}$$

$$= \begin{cases} +1 & \text{if } |\tilde{w}_{(l-1,i),(l,j)}| > \mathrm{WT} \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

There is inconsistency of notation: Above, $\tilde{w}_{(l-1,i),(l,j)}$ is used but in other places it is $\tilde{w}_{i,j}^{l,l-1}$. This occurs with other quantities too. We need to make notation consistent in whole document.

where A is the adjacency matrix of the binary graph $\mathcal{G}$(NN (Fig. 7), and we denote by D the depth of the ANN and $H(l)$ the number of units in layer l. The element $a_{(l-1,i),(l,j)}$ equals 1 if in $\mathcal{G}$(NN) an edge exists between the two nodes $(l-1, i), (l, j)$, 0 otherwise.

Another example of inconsistent notation is previous line with adjaceny matrix Note that after applying the graph-filtering procedure, some nodes can be disconnected from the resulting graph and the existence of a unique connected component can be guaranteed by the addition of one single edge. However, in our simulations, a unique connected component was always observed with some disconnected units whose weights were too weak in absolute values to be included in the resulting induced graph.

A.1.2. Graph statistics of interest

Due to the rigid structure of the graphs we consider - where edges only exist between two consecutive layers -, graph statistics that detect the graph shape, such as diameter or centrality measure are not expected to provide useful

additional information to compare the same architecture, trained in different strategies. Whereas, we focus on in-degree and out-degree of units in the hidden layers, defined in equation 11 and 15. These statistics respectively count the number of incoming and outgoing edges, capturing the amount of information flow through each unit (See Fig. 7). Since in our application, we only consider one hidden layer, we notate i a unit in the first layer $l = 1$. Thus, the value in minus out degree can be used to rank nodes in the hidden layer.

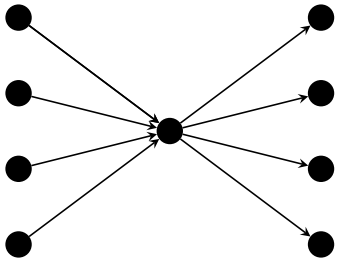
$$\deg^{\text{in}}(i) = \sum_{\substack{(l-1,j) \\ j\in\{1,\dots,H(l-1)\}}} a_{(l-1,j),(i)} \tag{14}$$

$$\deg^{\text{out}}(i) = \sum_{\substack{(l+1,j) \\ j\in\{1,\dots,H(l+1)\}}} a_{(i),(l+1,j)} \tag{15}$$

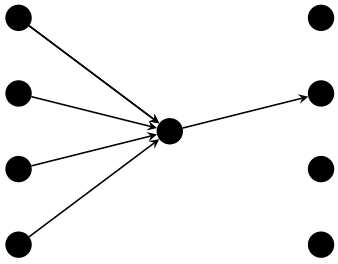
In particular, we consider the quantity

$$\mathcal{S}(i) = \deg^{in}(i) - \deg^{\text{out}}(i) \tag{16}$$

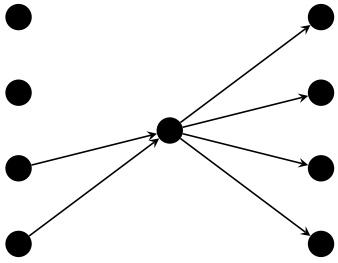
and we distinguish two types of nodes: nodes belonging to the interquartile range of $\mathcal{S}$ and nodes in the tail of $\mathcal{S}$. An example, with $\mathcal{S}$ having average in zero, can be visualized in Fig. 9



(a) Interquartile node, $\mathcal{S}(i) = 0$ with zero the median of $\mathcal{S}$



(b) Tail node which is channeling the information $\mathcal{S}(i) > Q3$



(c) Tail node which is irradiating the information $\mathcal{S}(i) < Q1$

Figure 9: Example of the two types of nodes

A.2. Material

A.2.1. Handwritten Digit Recognition

Despite being a universal task, digit handwriting is influenced by individual uniqueness in the formation and appearance of the digits (Jain and Ko [2008]). Educated humans gain expertise in recognizing handwritten digits all along their existence, from early school training, when such an ability is acquired, to adult life during which the ability is continuously refined to adapt to recognize distorted samples or more personal style (Legault et al. [1992]).

Training an artificial system to be competitive with humans in a handwritten digit recognition task is a fundamental step in human-machine interaction (LeCun et al. [1995], Ciregan et al. [2012], Kumar and Beniwal [2018], Niu and Suen [2012], Pashine et al. [2021]). In this field, the MNIST database Deng [2012] is widely used for benchmarking various image recognition algorithms. The database contains a total of 60,000 training and 10,000 test images of handwritten digits, each of which is 28x28 pixels in size, written by more than 500 different writers (LeCun et al. [1995]). The digits range from 0 to 9 as grayscale images.

Architecture and Learning Strategies

Different ANN architectures and algorithms of varying complexity have been proposed to tackle the classification of the MNIST dataset (Baldominos et al. [2019], Ciregan et al. [2012], Jarrett et al. [2009]. Due to the objective of our case of study, we used the MNIST database in a simple feedforward architecture defined as follows: an input layer of 784 units, a hidden layer of 512 units, and the output layer with 10 output neurons. Thus, the images were flattened before being fed into the ANNs. Comparable architectures are used in the literature for learning strategies evaluation (Goodfellow et al. [2013b], Kirkpatrick et al. [2017], Zenke et al. [2017], Lomonaco et al. [2021]).

We trained this architecture using different learning strategies (hyperparameters details are reported in the table 1):

- Sample Replay Lomonaco et al. [2021]
- GDumb Prabhu et al. [2020]
- SI Zenke et al. [2017]
- EWC Kirkpatrick et al. [2017]
- LwF Li and Hoiem [2017]
- Cumulative
- Finetune

For each architecture and learning strategy, we tested different randomly selected orders. In Table 2, we report for

Table 1
Hyperparameters of the considered learning strategies settings. * See Kingma and Ba [2014], *SGD = Stochastic Gradient Descent.

Strategy	Hyperparameters			
	Units in [input, hidden, output]	Activation functions	Optimizer	Loss function
Cumulative	[784, 512, 10]	[relu, softmax]	Adam*	Cross-entropy
Finetune	[784, 512, 10]	[relu, softmax]	Adam	Cross-entropy
Sample Replay	[784, 512, 10]	[relu, softmax]	Adam	Cross-entropy
GDumb	[784, 512, 10]	[relu, softmax]	Adam	Cross-entropy
EWC	[784, 512, 10]	[relu, softmax]	Adam	Cross-entropy
LwF	[784, 512, 10]	[relu, softmax]	Adam	Cross-entropy
SI	[784, 512, 10]	[relu, softmax]	Adam	Cross-entropy

Table 2
List of the trained configuration results on the handwritten digit recognition database MNIST. Acc: global accuracy after training on all sessions.

Order	Strategy	Acc.	*stab.*	*plas.*
(A) 1036457928	Sample Replay	0.80	0.77	0.90
	GDumb	0.23	0.00	0.32
	EWC	0.12	0.14	1.00
	LwF	0.15	0.21	0.99
	SI	0.13	0.19	1.00
	Finetune	0.11	0.08	1.00
	Cumulative	0.72	1.00	0.59
(B) 2813047569	Sample Replay	0.60	0.94	0.55
	GDumb	0.54	0.42	0.63
	EWC	0.31	0.03	0.99
	LwF	0.46	0.11	0.97
	SI	0.45	0.12	0.98
	Finetune	0.14	0.00	1.00
	Cumulative	0.68	1.00	0.59
(C) 1204673985	Sample Replay	0.82	0.56	0.88
	GDumb	0.41	0.00	0.53
	EWC	0.15	0.00	1.00
	LwF	0.23	0.00	0.98
	SI	0.22	0.00	0.99
	Finetune	0.11	0.00	1.00
	Cumulative	0.80	1.00	0.65
(D) 4816203957	S. Sample Replay	0.68	0.91	0.72
	GDumb	0.48	0.24	0.57
	EWC	0.25	0.00	0.99
	LwF	0.37	0.07	0.98
	SI	0.32	0.02	0.99
	Finetune	0.17	0.00	0.99
	Cumulative	0.73	1.00	0.62
(E) 6031925487	Sample Replay	0.78	1.01	0.80
	GDumb	0.46	0.64	0.53
	EWC	0.26	0.18	0.99
	LwF	0.31	0.36	0.98
	SI	0.29	0.31	0.99
	Finetune	0.17	0.00	1.00
	Cumulative	0.70	1.02	0.52
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
(F) 941752386	Sample Replay	0.73	1.00	0.80
	GDumb	0.40	0.84	0.42
	EWC	0.20	0.43	0.99
	LwF	0.26	0.68	0.98
	SI	0.24	0.53	0.99
	Finetune	0.12	0.15	1.00
	Cumulative	0.71	1.00	0.45
(G) 9480712653	Sample Replay	0.70	0.89	0.74
	GDumb	0.44	0.00	0.55
	EWC	0.24	0.00	0.99
	LwF	0.31	0.00	0.91
	SI	0.29	0.00	0.96
	Finetune	0.13	0.00	1.00
	Cumulative	0.65	1.01	0.55
(H) 6180534972	Sample Replay	0.76	0.97	0.75
	GDumb	0.42	0.89	0.44
	EWC	0.18	0.46	0.99
	LwF	0.30	0.78	0.97
	SI	0.25	0.72	0.99
	Finetune	0.13	0.14	1.00
	Cumulative	0.70	1.00	0.60
(I) 2784903165	Sample Replay	0.64	0.97	0.70
	GDumb	0.47	0.62	0.56
	EWC	0.23	0.38	0.99
	LwF	0.37	0.52	0.91
	SI	0.29	0.52	0.96
	Finetune	0.12	0.10	1.00
	Cumulative	0.76	0.98	0.70
(J) 2536790418	Sample Replay	0.69	0.98	0.73
	GDumb	0.47	0.69	0.51
	EWC	0.21	0.17	0.98
	LwF	0.33	0.49	0.95
	SI	0.24	0.28	0.98
	Finetune	0.14	0.00	1.00
	Cumulative	0.83	0.99	0.68

our simulations, the achieved results in *stability*, *plasticity*, last global accuracy ($\frac{\text{number of correctly predicted samples}}{\text{total number of samples}}$) and Ω_{all} ($\Omega_{\mathbf{all}} = \frac{1}{T-1}\sum_{i=2}^{T}\frac{\alpha_{\text{all,i}}}{\alpha_{\text{ideal}}}$ as defined in Kemker et al. [2018]).

A.2.2. Face Emotion Recognition

In human social interaction, the ability to identify other beings feeling and emotions is crucial, particularly to adapting the individual's behavior. Emotion recognition is mainly

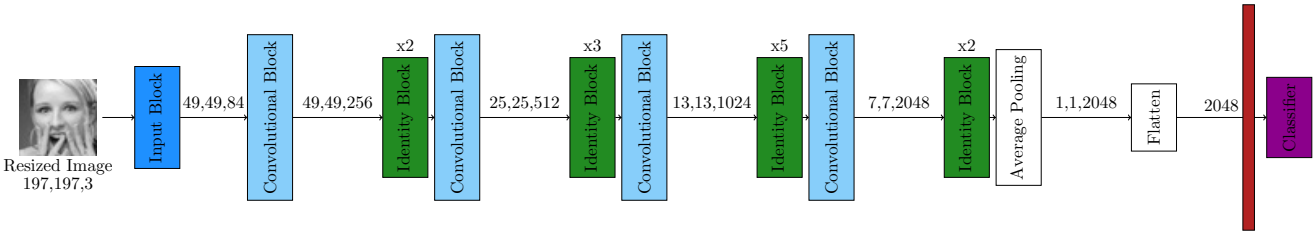


Figure 10: ResNet50 model architecture. The features vector taken as the input of the last fully connected layers is extracted in correspondence of the red box. The Convolutional Blocks extract features changing the input dimensions. The Identity Blocks extract features without changing the input dimensions.

achieved, but not exclusively, by decoding non-verbal information and in particular facial expressions. Many social cognition studies focus on understanding emotion recognition in humans, for instance detecting specific region contributions (in particular in the amygdala Adolphs et al. [1994, 1995], Calder [1996], Breiter et al. [1996], Morris et al. [1996], Davis and Whalen [2001], Pessoa and Adolphs [2010], Anderson and Phelps [2000]) or defining functional activation at different emotional faces processing (Fusar-Poli et al. [2009], Gómez et al. [2020], Liao et al. [2021], Underwood et al. [2021]).

Automatic systems and ANNs for facial expression recognition have also been introduced (Kumari et al. [2015], Mehta et al. [2018], Li and Deng [2020]). Available databases mainly cover the six basic emotions: Anger, Fear, Sadness, Disgust, Surprise and Happiness (Ekman [1992a,b], Ekman and Friesen [1978]) and a Neutral emotional state. Here, we considered the Fer+ databases (Goodfellow et al. [2013a], Barsoum et al. [2016]). This database contains 35,685 grayscale 48x48 pixels images with all the basic emotions and covering all ages, gender, and ethnicity, the labels are provided by 10 crowd taggers.

Architecture and Learning Strategies

We follow the work of Mainsant et al. [2021] which introduces an ANN allowing continual learning without the requirement of neurogenesis, nor the necessity of an oracle about learned or data to be learned, as well as data privacy issues for facial emotion recognition.
In a preliminary step, we performed a feature extraction employing a pre-trained ResNet50 model provided by Wang et al. [2018]. A visualization of ResNet50 architecture can be found in Fig.10. The feature vectors used as input of our NN architectures were obtained after the flatten operation of ResNet (indicated by a red rectangle in Fig.10).

The 2048-feature vectors are then fed into the following fixed architecture:

- input layer of 2048 units corresponding to the size of features extracted from images;
- hidden layer with 1000 neurons;
- output layer.

We evaluated Dream Net together with the other learning strategies (as listed in Table 3). Note that the methods do not entirely share the same architecture: indeed Dream Net requires an output layer composed of several neurons corresponding to the input (Auto-associative or Auto-encoder part) and several neurons corresponding to the number of classes (Hetero-associative or part). While this part is used both for training and evaluation of the model, the Dream Net activation network at rest is obtained by discarding all the units coming from the auto-associative part and their related edges, so that only the part of the ANN directly involved in the classification task is retained. This guarantees a fair comparison with the other learning strategies and architectures.

Dream Net architecture can be divided into three phases:

1. The *Learning Net* at each learning session, learns real features from the class of the session and pseudo-features from the previously learned classes.
2. The *Learning Net* at the end of the learning session transfers its weights to *Memory Net*
3. *Memory Net* captures the learned function using a re-injection sampling procedure. The re-injection sampling procedure consists of the following steps: inject a random noise input vector and re-inject the replication vector obtained at the output of the auto-associative part of *Memory Net* at its input and so on. At each re-injection, Auto and Hetero associative outputs of *Memory Net* are conserved to create pseudo-examples. After several re-injection, a pseudo-examples database is obtained that contains pseudo-features and corresponding pseudo-labels obtained after each re-injection (data from the first inference is not kept).

A visualization scheme of the Dream Net learning procedure can be found in Figure 11.

The architecture was sequentially trained with different learning strategies and in different learning sessions. We tested 7 choices of emotion orders so that each class is learned in the first position, followed by random order choice.

The different learning configurations are listed in table 4, with the learning orders notated by A: angry, D: disgust, F: fear, H: happy, S: sad, Su: surprise, and N: neutral.

Table 3
Hyperparameters of the considered model settings. * See Kingma and Ba [2014], *SGD = Stochastic Gradient Descent.

	Strategy	Hyperparameters		
	Units in [input, hidden, output]	Activation functions	Optimizer	Loss function
Cumulative*	[2048, 1000, 2055]	[relu, sigmoid]	Adam*	Binary Cross-entropy
Finetune*	[2048, 1000, 2055]	[relu, sigmoid]	Adam	Binary Cross-entropy
Dream Net	[2048, 1000, 2055]	[relu, sigmoid]	Adam	Binary Cross-entropy
Cumulative	[2048, 1000, 7]	[relu, sigmoid]	Adam	Binary Cross-entropy
GDumb	[2048, 1000, 7]	[relu, sigmoid]	Adam	Binary Cross-entropy
Sample Replay	[2048, 1000, 7]	[relu, sigmoid]	Adam	Binary Cross-entropy
EWC	[2048, 1000, 7]	[relu, sigmoid]	Adam	Binary Cross-entropy
LwF	[2048, 1000, 7]	[relu, sigmoid]	Adam	Binary Cross-entropy
SI	[2048, 1000, 7]	[relu, sigmoid]	Adam	Binary Cross-entropy
Finetune	[2048, 1000, 7]	[relu, sigmoid]	Adam	Binary Cross-entropy

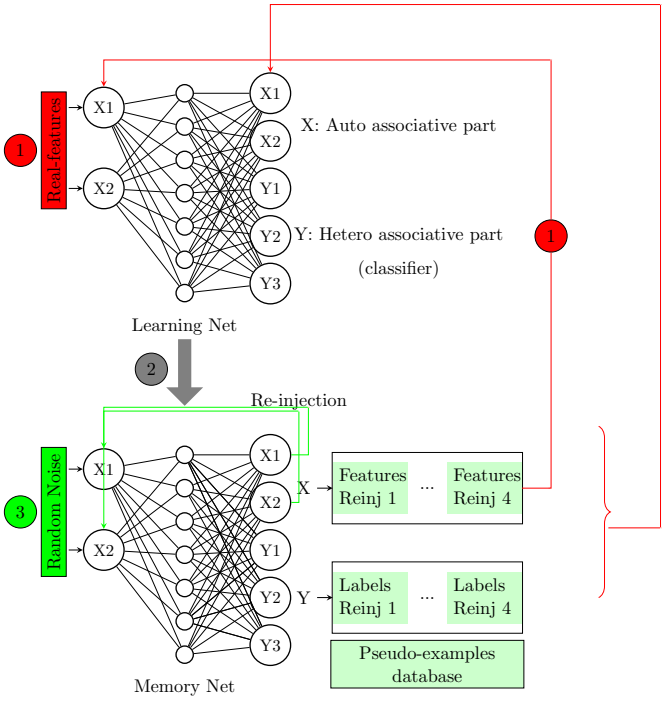


Figure 11: Architecture Scheme of Dream Net model. The Dream Net model learning procedure can be divided into 3 phases indicated as 1, 2, and 3.

The model setting performance evaluation is based on the last global accuracy and on the metrics proposed in Kemker et al. [2018].

B. Experimental Details

Following the standard human brain functional connectivity analysis framework, we performed two types of experiments on the induced graph $\mathcal{G}(NN)$ of various NNs: a global setting recognition and a nodal role identification. For the former, we assumed that the induced graphs of a NN contained enough information to identify its learning settings. For the latter, we associated nodal graph statistics with a functional property of the artificial system. In particular, we performed a network surgery, by removing hidden units according to their statistics value. Thus, we evaluated the *stability* and the *plasticity* performance: if removing the units had a strong negative effect, thus the units are considered as critical. In particular, we defined a set of units as stability-critical (resp. plasticity) if by pruning all its units, we observed negative changes in terms of stability (resp. plasticity) performance. If the evaluation performance in the pruned version is increased, then we defined the unit set as stability/plasticity-inhibitorial.

B.1. Model setting recognition

Similarly to human brain studies where functional connectivity allows to discriminate across brain states, the first objective of our analysis was to demonstrate that a graph-based connectivity analysis of ANNs could discriminate architectures according to different learning strategies. Particularly, we explored the possibility of correctly identifying models whose learning was affected by catastrophic forgetting.

Graph features extraction. To this extent, we considered the induced graphs of each configuration in different learning settings. We determined the threshold to guarantee in and out degrees statistics followed a similar distribution (an example is shown in Fig. 12).

In T learning sessions (corresponding to the number of classes to learn), we determined the induced graph at the end of each session, thus for each configuration (an architecture, a learning strategy, a database, and the learning order), we extract $T+1$ simple features as follows. First, we define the Maximum Unit Change in Equation 17.

$$\text{Maximum Unit Change} = \max_{\substack{i\in\{1,\dots,H(l)\}\\ t\in\{1,\dots,T\}}} \left| S_t(i) - S_{t-1}(i) \right| \quad (17)$$

Since we only have one hidden layer, is it better to simply get rid of $H(l)$ notation? where $H(l)$ is the number of units in layer l. Next, we determined the number of units in the tail of S_t at each learning session, given by v_t in Equation 18.

$$v_t = \sum_{i\in\{1,\dots,H(l)\}} \chi_{\{j|S_t(j)<Q1_t\}}(i) + \chi_{\{j|S_t(j)>Q3_t\}}(i) \quad (18)$$

Table 4
List of the trained configuration results on the face emotion recognition database Fer+. Acc: global accuracy after training on all sessions.

Order	Strategy	Acc.	*stab.*	*plas.*
(A) ADFHSSuN	Cumulative	0.99	1.01	0.67
	Cumulative*	0.76	1.03	0.81
	Dream Net	0.71	0.81	0.84
	Sample Replay	0.57	0.52	0.95
	GDumb	0.61	0.57	0.79
	EWC	0.10	1.15	0.00
	LwF	0.34	0.00	0.83
	SI	0.34	0.00	1.00
	Finetune*	0.02	0.00	1.00
	Finetune	0.032	0.19	0.17
(B) DAHSNSuF	Cumulative	0.76	1.00	0.40
	Cumulative*	0.76	1.03	0.84
	Dream Net	0.77	1.00	0.82
	Sample Replay	0.54	0.54	0.26
	GDumb	0.52	0.53	0.25
	EWC	0.099	0.00	0.17
	LwF	0.34	0.19	0.17
	SI	0.34	0.19	0.17
	Finetune	0.032	0.00	1.00
	Finetune*	0.34	0.00	1.00
(C) FHSuDNSA	Cumulative	0.78	0.86	0.39
	Cumulative*	0.76	1.56	0.81
	Dream Net	0.51	1.31	0.32
	Sample Replay	0.57	0.68	0.14
	GDumb	0.62	0.81	0.27
	EWC	0.099	0.00	0.17
	LwF	0.34	0.22	0.17
	SI	0.34	0.22	0.00
	Finetune	0.099	0.00	1.00
	Finetune*	0.099	0.00	1.00
(D) HADNSFSu	Cumulative	0.76	1.01	0.67
	Cumulative*	0.76	0.99	0.81
	Dream Net	0.71	1.00	0.86
	Sample Replay	0.56	0.54	0.35
	GDumb	0.43	1.00	0.40
	EWC	0.099	0.00	0.17
	LwF	0.34	0.24	0.33
	SI	0.34	0.24	0.17
	Finetune	0.12	0.00	1.00
	Finetune*	0.34	0.00	1.00
⋮	⋮	⋮	⋮	⋮
(E) SNDFSuAH	Cumulative	0.77	0.52	0.50
	Cumulative*	0.77	1.00	0.85
	Dream Net	0.71	0.99	0.90
	Sample Replay	0.37	0.60	0.33
	GDumb	0.37	0.80	0.25
	EWC	0.099	0.00	0.17
	LwF	0.34	0.23	0.17
	SI	0.34	0.23	0.00
	Finetune	0.26	0.00	1.00
	Finetune*	0.36	0.00	1.00
(F) SuDFNHAS	Cumulative	0.76	0.33	0.57
	Cumulative*	0.76	1.00	0.86
	Dream Net	0.71	0.93	0.92
	Sample Replay	0.41	0.68	0.30
	GDumb	0.34	0.66	0.41
	EWC	0.099	0.00	0.17
	LwF	0.34	0.20	0.17
	SI	0.34	0.20	0.00
	Finetune	0.13	0.00	1.00
	Finetune*	0.13	0.00	1.00
(G) NDSuFHSA	Cumulative	0.76	0.77	0.68
	Cumulative*	0.76	1.00	0.83
	Dream Net	0.56	0.79	0.58
	Sample Replay	0.34	0.84	0.20
	GDumb	0.34	0.77	0.43
	EWC	0.099	0.51	0.17
	LwF	0.34	0.61	0.00
	SI	0.34	0.20	0.17
	Finetune	0.098	0.00	1.00
	Finetune*	0.098	0.00	1.00

$Q1_t, Q3_t$ are the first and third quartiles of $S_t(j)$ distribution. In general, we can expect $S_t(i)$ to be positive. Yet, a node hosting a negative flow may exist. It corresponds to a neuron receiving little information and spreading it irrespective to all output units. Vice versa, a very high $S_t(i)$ value represents a neuron aggregating multiple information into a few output units.

Then, the extracted graph feature Maximum Unit Change represents the maximum consecutive change in the flow of one hidden neuron across task learning t . Each v_t quantifies the number of units in the hidden layer whose flow does not belong to its interquartile range.

Clustering. Given the $T + 1$ features per configuration, we determined the principal components and perform a DBSCAN or a K-Means Clustering algorithm in the reduced space. Thus, we evaluated the consensus score by computing the adjusted rand index where the true labels are given by the learning strategies or by two classes *affected by catastrophic forgetting, not affected* which corresponds to the configurations where $stability < 0.5, stability \geq 0.5$.

B.2. Network Surgery: Nodal Role Identification

We are interested in determining the behavior of hidden units according to their value in $S_t(i)$. In particular, for each model and configuration NN, we notate the ordered list of trained networks as $\mathbb{NN} = (\mathrm{NN}^0, \mathrm{NN}^1, \ldots, \mathrm{NN}^t, \ldots, \mathrm{NN}^T)$.

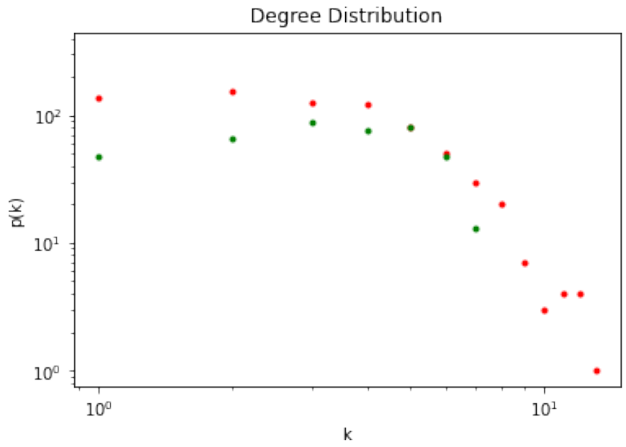


Figure 12: Example of in (green) and out (red) degree distributions in the induced graph of an activation network at rest.

At each learning session t, we extract the associated induced graph and compute for each hidden unit i the statistics $S_t(i)$. Thus, we compute the first and third quartiles $Q1_t, Q3_t$. We notate D the depth of the NN and H_A the number of units in the auto-associative part of the output layer. In our experiments, we have the parameters: $D = 3, H_A = 0, H(l = \text{hidden layer}) = 574$ in the handwritten digit recognition task and $D = 3, H_A = 0/2048, H(l = \text{hidden layer}) = 1000$ in the face emotion recognition task, with $H_A = 2048$ for the Dream Net architecture.

We propose the following definitions. First, a version $\mathbb{NN}^\tau = (\text{NN}^{0,\tau}, \text{NN}^{1,\tau}, \cdots \text{NN}^{T,\tau})$ which only preserves the weights of units outside the interquartile in-out degree interval, namely in the *tail* τ.

$\text{NN}^{t,\tau}$ is thus, the neural network with all synaptic weights equal to the not-pruned version at session t NN^t ($\mathbf{w}_{i,j}^{l,l+1}(t)$), except for the synaptic weights of the hidden layer units i whose $S_t(i)$ is in the interquartile which is fixed to zero, as follows:

$$\mathbf{w}_{i,j}^{(\tau)l,l+1} = \begin{cases} 0 & Q1_t < S_t(i) < Q3_t \\ \mathbf{w}_{i,j}^{l,l+1}(t) & \text{otherwise} \end{cases} . \tag{19}$$

Second, a complementary version of the previous one, which discards the weights of units in the tail. We notate this version $\mathbb{NN}^{\bar{\tau}} = (\text{NN}^{0,\bar{\tau}}, \text{NN}^{1,\bar{\tau}}, \cdots, \text{NN}^{T,\bar{\tau}})$.

$\text{NN}^{t,\bar{\tau}}$is the neural network with synaptic weights

$$\mathbf{w}_{i,j}^{(\bar{\tau})l,l+1} = \begin{cases} \mathbf{w}_{i,j}^{l,l+1}(t) & Q1_t < S_t(i) < Q3_t \\ 0 & \text{otherwise} \end{cases} , \tag{20}$$

with $\mathbf{w}_{i,j}^{l,l+1}(t)$ the synaptic weights of the standard not-pruned trained networks NN^t. A schematic visualization of these pruned version is shown in Fig. 13.

C. Related Works

Complex network tools have been used for the characterization of trained and untrained ANNs. In (La Malfa et al. [2021, 2022]), the authors apply complex network theory to deep ANNs, having different layers configuration (fully connected, convolution layer, etc.). Their ANNs characterization is based on metrics distribution over weighted directed graphs. As an alternative, our approach is inspired by brain network analysis (Richiardi et al. [2013], Fornito et al. [2013], De Vico Fallani et al. [2017], and based on binary networks. By focusing on the degree nodal statistics, we introduce a fine characterization at the neuron level.

A motif discovery process is proposed in (Zambra et al. [2020]), where the authors characterize multi-layer perceptron learning by detecting different patterns of connection of groups of four or five units. Their method is used to compare different weights initialization in multi-class classification. While the proposed framework has promising results, the motif search can become computationally expensive while the dimension of ANN increases (Masoudi-Nejad et al. [2012], Patra and Mohapatra [2020]). In their application, the ANN size appears quite small (30-20 units and 3 layers). We introduce a less costly framework based on simple and intuitive metrics.

Next, the work (Scabini and Bruno [2021]) extends complex network techniques to detect different neuron types and to relate their presence to the performance of fully connected neural networks. Their framework considers only weighted graphs and an Offline learning setting.

A different graph model definition is proposed in (Hanczar et al. [2020]), where the ANN decision process for a binary classification task is based on the definition of a *relevance network* per class, obtained through the computation of the layer-wise relevance propagation (LRP) score (Bach et al. [2015]). While LRP has been defined and it is mainly used to determine input features contributing to the final classification, the authors originally propose to use the class relevance network as a human-understandable decision process. Their approach associates each unit with the final decision, providing a human-understandable decision process but requires expert knowledge insertion.

Finally, a new approach has been proposed in (Corneanu et al. [2019]) by the definition of a *functional network* obtained by the computation of Pearson correlation among activation units. Multiple instances of functional networks across training epochs are then compared using topological metrics in order to assess the model evolution during the learning process. In their approach, the ANN model is treated as a structural network and its weights and architecture are not considered.

D. Supplementary Results

D.1. Clustering Results

Figures 14, 15 compare the true configuration labels with the assigned clusters in the reduced spaces. With the extracted graph features, it is possible to detect the occurrence of the catastrophic forgetting phenomenon and in the smallest architecture, we can detect the different learning strategies.

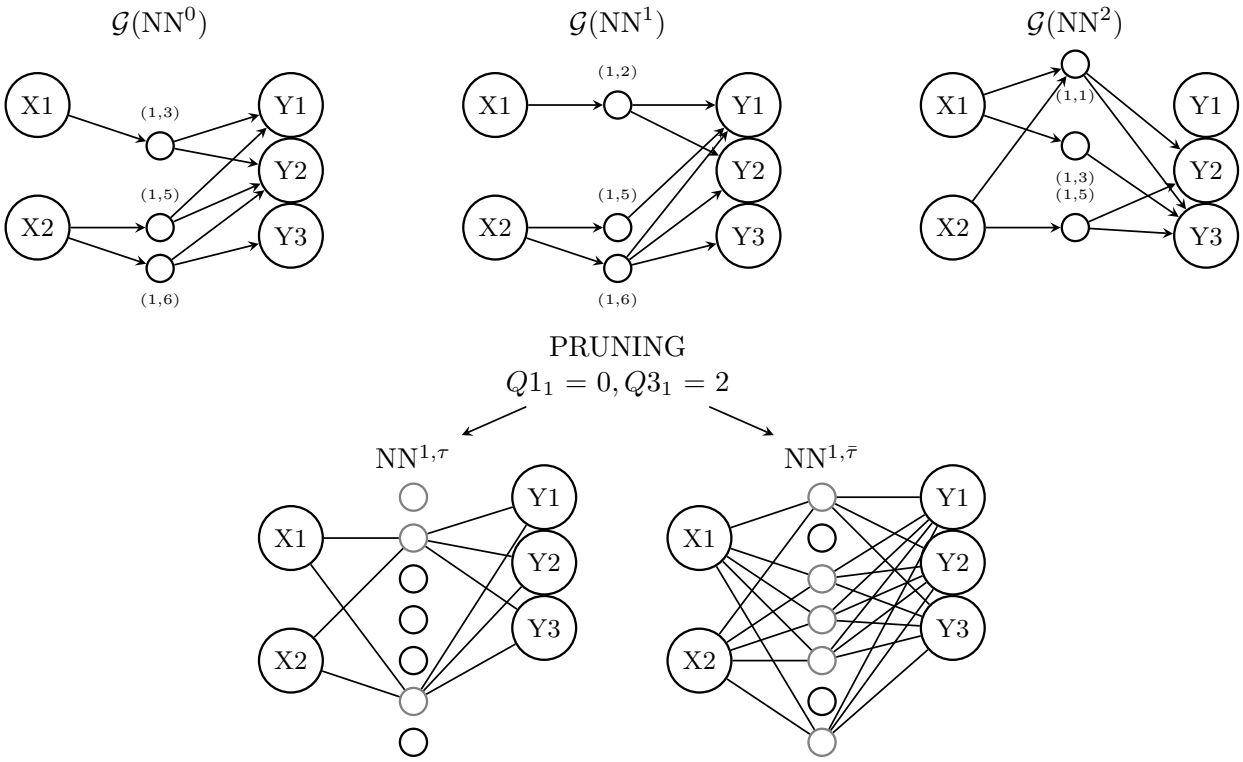


Figure 13: Schematic visualization of the pruning versions. Gray units correspond to the not-pruned ones.

D.2. Network Surgery

Figures 16 and 17 show the network surgery results in all the considered configurations.

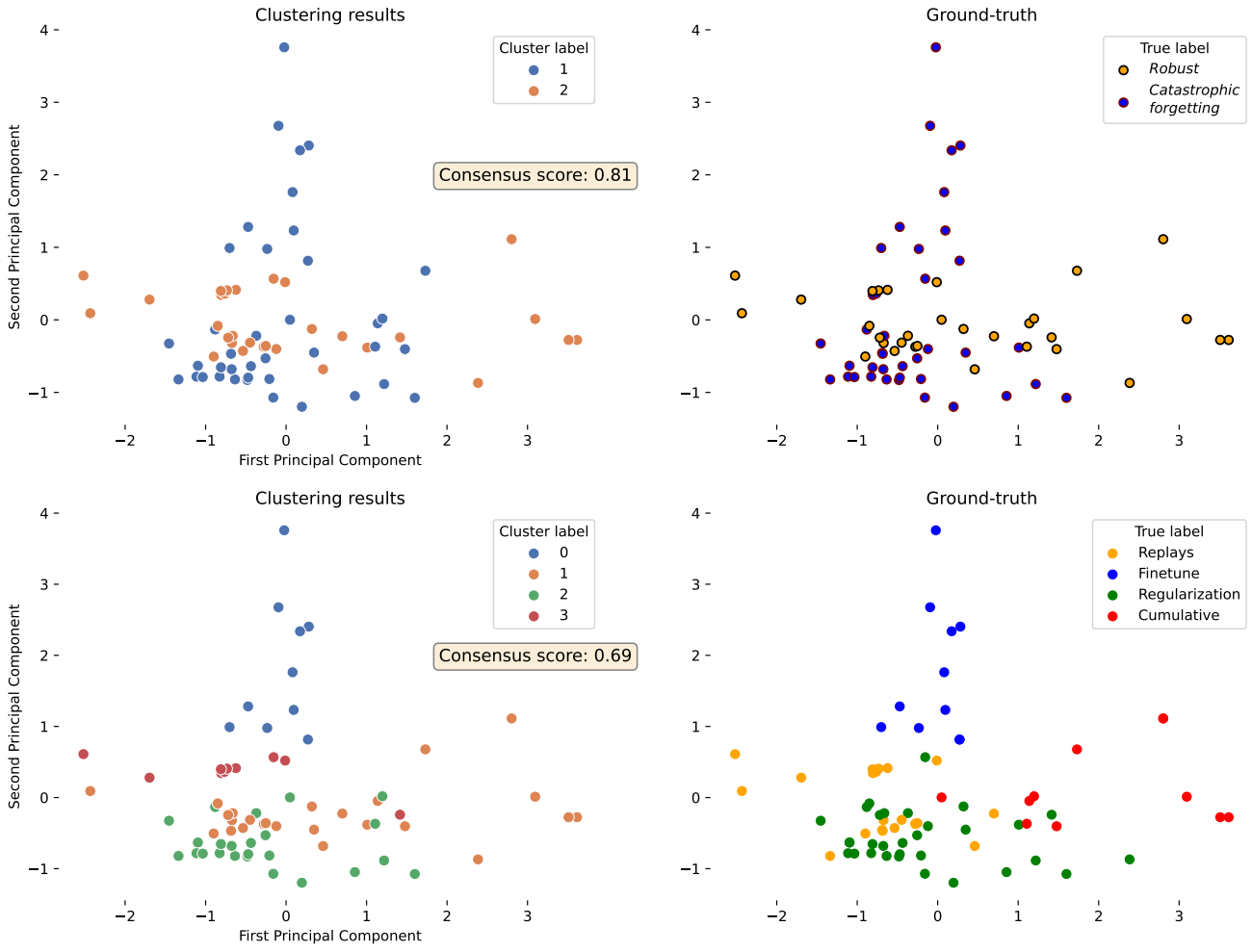


Figure 14: Visualization of the cluster labels (Left) obtained by applying KMeans (Top: K=2, Bottom: K=4) in the reduced space of the Handwritten Recognition task and their ground truth (Right).

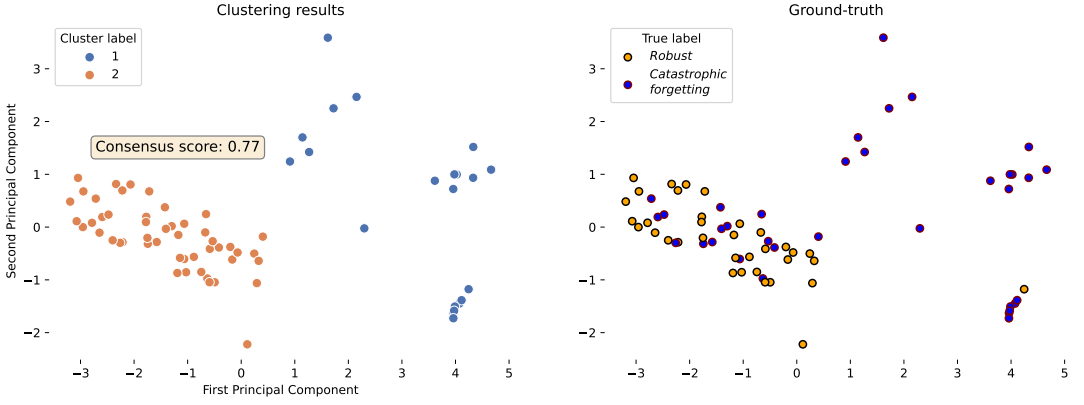


Figure 15: Visualization of the cluster labels (Left) obtained by applying KMeans with K=2 in the reduced space of the Face Emotion Recognition task and their ground truth (Right).

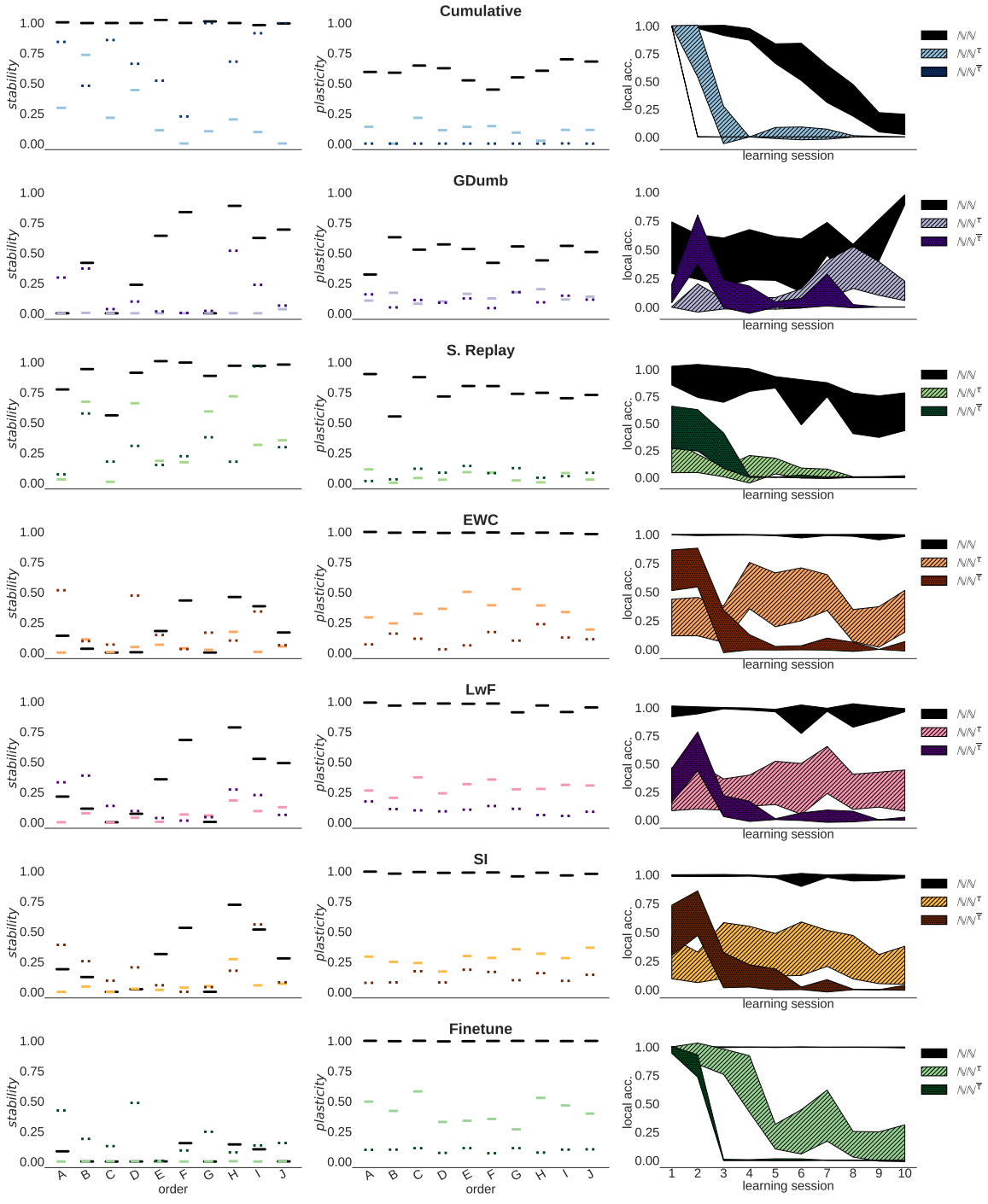


Figure 16: Network surgery results for all the considered orders and strategies on MNIST task. See the legend of Figure 5.

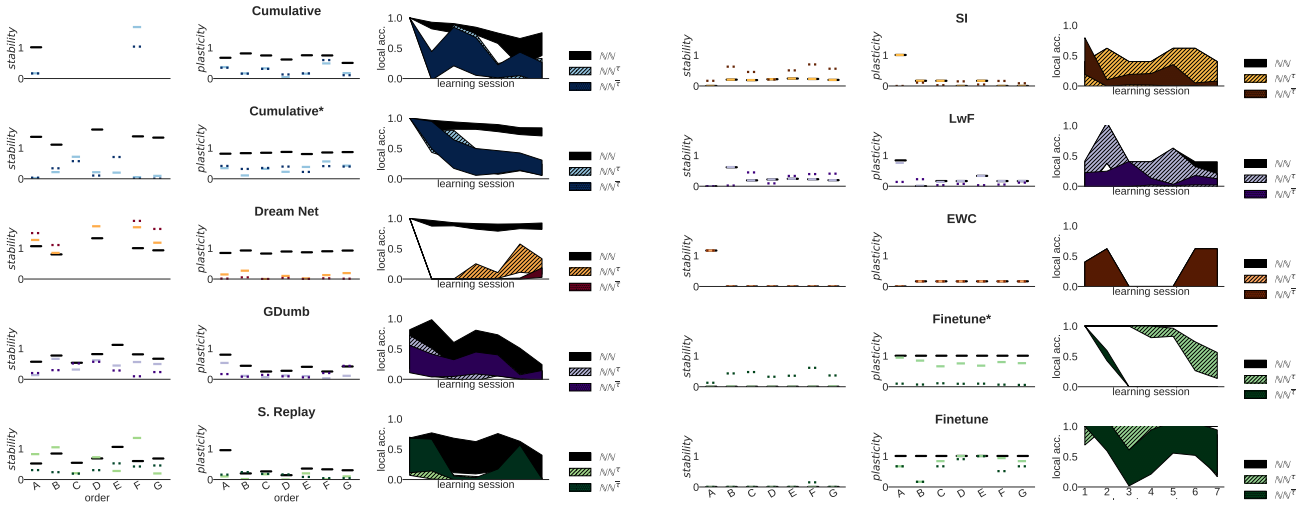


Figure 17: *Stability* (Left) and *Plasticity* (Center) performances on emblematic configurations on face emotion recognition task in their pruned versions. For all different orders (see Tab. 4) of learning sessions, we report the performance of the standard model $\mathbb{NN}$ in plain black line. The change in the performance when the model is pruned is reported for the two different pruning versions and colored. (Right): average local accuracy performances at different learning sessions. For the standard $\mathbb{NN}$ and the pruned models $\mathbb{NN}^{\tau}, \mathbb{NN}^{\bar{\tau}}$ the accuracy for the last learned class across learning sessions is reported with its confidence interval for all studied configurations.